Distributed Localization of Active Transmitters in a Wireless Sensor Network

THESIS

Oba L. Vincent, 2nd Lieutenant, USAF

AFIT/GE/ENG/12-41

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/12-41

Distributed Localization of Active Transmitters in a Wireless Sensor Network

THESIS

Presented to the Faculty

Department of Electrical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Masters of Science in Electrical Engineering

Oba L. Vincent, B.S.E.E.

2nd Lieutenant, USAF

March 2012

AFIT/GE/ENG/12-41

Distributed Localization of Active Transmitters in a Wireless Sensor Network

Oba L. Vincent, B.S.E.E.

2nd Lieutenant, USAF

Approved:

/signed/	29 Feb 2012
Maj. Mark D. Silvius, Ph.D. (Chairman)	date
/signed/	29 Feb 2012
Dr. Richard K. Martin (Member)	date
/signed/	29 Feb 2012
Dr. Kenneth M. Hokinson (Member)	date

AFIT/GE/ENG/12-41

Abstract

In today's military environment, emphasis has been placed on bandwidth efficiency and total use of the available spectrum. Current communication standards divide the spectrum into several different frequency bands, all of which are assigned to one or multiple primary users. Cognitive Radio utilizes potential white spaces that exist between currently defined channels, or in the time between channel communications. One under-explored dimension of white space exploration is spatial. If a frequency band is being used in one region, it may be underutilized, or not occupied at all in another. Using an active localization method can allow for the spatial white spaces to be discovered. Trying to spatially map all of the frequencies in a large area would be come very computationally intensive, and may even be impractical using modern centralized methods. Applying a distributed method and the concepts discussed in Wireless Distributed Computing to the problem can be scaled onto many small wireless sensors and could improve the measuring system's effectiveness. For a bandwidth contested environment that must be spectrally mapped, three metrics stand out as critical: Accuracy, Power Consumption, and Latency. All of these metrics must be explored and measured to determine which method could be most effectively applied to the spectral mapping of a spatial environment.

Acknowledgements

This research efforts was made possible with the love and support of my wife and family, without whom I would have never been able to finish this work. My wife has proofread this entire document, and with out her I would still be looking for spelling mistakes.

I would also like to thank all of my professors from both my undergrad and graduate work. The knowledge and engineering concepts they have provided my has taught me to persevere through even the most difficult parts of this thesis.

Finally I would like to the Maj Silvius for the help that he gave me in solving the many problems that I encountered and providing guidance when I was unsure of which path to take next. His guidance and leadership has allowed this topic to develop into my thesis.

Oba L. Vincent

Table of Contents

Page

List of Figures

Figure | Page

Figure Page

List of Tables

Distributed Localization of Active Transmitters in a Wireless Sensor Network

I. Introduction

As technology continues to improve, the ways in which military forces fight wars must continue to evolve. Modern advances in radio communication are critical to expanding communication capabilities. Radios today have the ability to operate on a wide range of frequencies. They can be designed to sense frequency bands for occupied channels, to find friendly radios with which to connect, or to find enemy transmissions that should be intercepted, jammed, or even geolocated. Warner J.A. Dahm, the Chief Scientist of the Air Force, and his office, published a document demonstrating projected capabilities over the next 20 years, resulting from new developing technologies being. One primary category is "Processing-Enabled Intelligent ISR Sensors," which are information sensors with inherent processing capabilities [4]. Additionally, an emphasis on "Frequency Agile Spectrum Utilization," also described by the Chief Scientist of the Air Force, is focused on distributed methods to measure and map the available spectrum, and on using this knowledge to more efficiently operate in that radio environment. With an end goal to achieve these two research areas it is important to develop techniques and methods to rapidly employ this technology as it evolves. To prepare and develop future technology, improved bandwidth performance and power performance are required, while maintaining the accuracy of the current

system. This research aims to explore the potential improvement of localization systems using Received Signal Strength by using distributed methods for data collection and processing.

1.1 Motivation

In war, information can be a military force's most valuable resource. For hundreds of years military forces have succeeded and failed based on information about enemy troop sizes, troop movements, and the morale of those forces. In today's battlefield, information is constantly being gained and updated from sensors spread throughout the entire Area of Responsibility (AOR). Increased sensor usage brings about issues with bandwidth and power management at each of these sensor nodes. Distributed Processing can greatly improve both these as well as available processing power. The bandwidth problem associated with the battlefield has been a well documented. According to Anthony H. Cordesman, "There are questions to how the United States will solve the bandwidth issues discovered in Afghanistan" [5]. It is therefore imperative that all systems be designed for bandwidth efficiency to reduce their bandwidth footprint, this reduction must also maintain accuracy. Additionally, power must be used more efficiently on small devices in order to extend the battery life of sensors. The localization process must be kept as simple as possible to allow for processing on small sensors with minimal power.

We can apply distributed processing to the many different tasks a small wireless sensor array may need to perform. Two such useful tasks are spectrum sensing and

localization. By combining these two tasks it is possible to make a map that shows the usage of the spectrum in 3D space. A spectrum usage map would contain valuable information for tracking radio locations, cognitively jamming enemy communication channels, and finding channels for friendly communication. The map could also be used to locate key enemy Command and Control (C2) facilities, which may use higher bandwidth. These facilities could then be selected as potential targets for conventional weapons or unconventional jamming attacks.

Localization methods with minimal power and bandwidth footprints, could be applied to several problems in the current operating environment, so as to provide a tactical advantage to users on the ground. First, they can be used to detect interference between different radio networks, created by multiple users. By localizing a transmitter, and then predicting its transmission, communications interference can be predicted and prevented. Second, using a localization network can also allow for asset tracking on a base. Tracking an active RFID tag or a human carrying a cell phone can be used to monitor for safety and security on a flight line. Third, detection along a fence line can allow for a low cost intrusion detection system, and can allow for enemy positions to be monitored.

1.1.1 Scenario. A wireless sensor network used for scanning radio spectrum can be utilized in a number of scenarios. One example would be during the initial deployment to foreign country. By establishing a perimeter, and placing sensors around it, useful spectrum usage information can be gathered. It could be used

to detect civilian transmitters like television and radio stations. By locating this type of transmitter the US forces can avoid these frequency locations and prevent interfering with the civilian populace. The second use would be for detection of enemy transmitters, such as a enemy radio or cell phone. This type of device could be tracked and monitored for potential security breaches. The third use would better allow a radio spectrum map to be developed of the current radio environment. The map would allow for prevention of friendly interference from multiple users utilizing the same channel. For this concept to become a reality, advancements must be made in the bandwidth efficiency, power efficiency, latency, and accuracy of wireless localization networks.

1.2 Problem Statement

Wireless Distributed Computing allows for the processing of complex problems and large collections of data using wireless sensors. Can Received Signal Strength localization methods be used with Wireless Distributed Computing to create an effective method of localization? Also, how does this method of localization compare with the centralized approach that has been previously explored? Which method is better in terms of Accuracy, Power Consumption, Bandwidth, and Latency? What situations allow for the best performance for each?

1.3 Background

The research in this thesis combines several research areas. These areas are briefly described. Figure 1.1 shows how these disciplines are combined in this research to develop the methods used in the thesis.

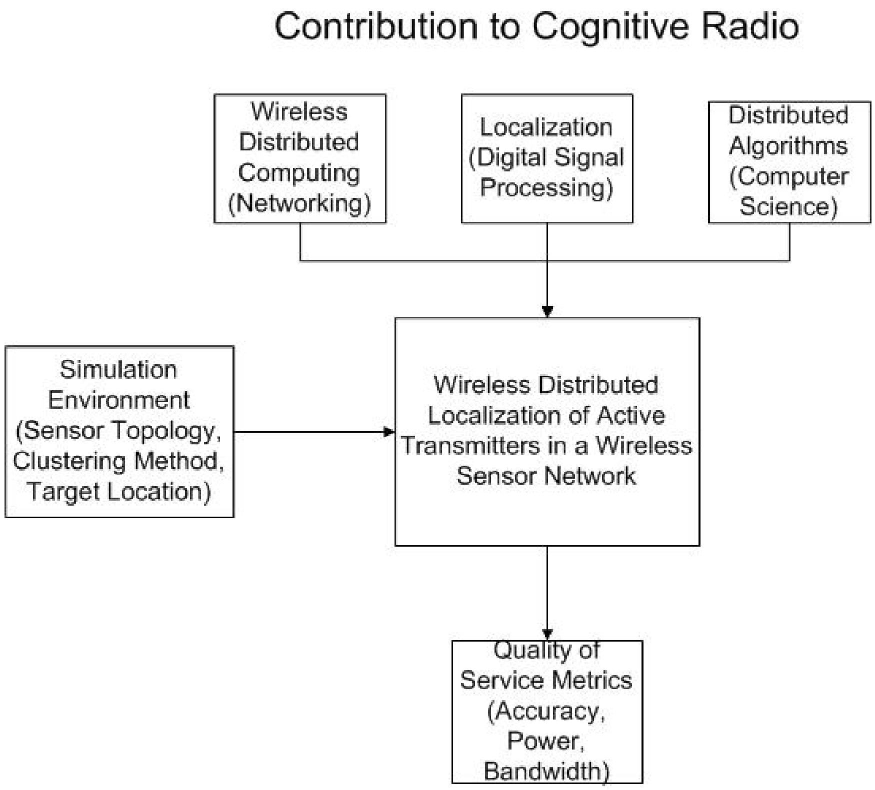

Figure 1.1: Thesis Concept Map

1.3.1 Cognitive Radio and Spectrum Sensing. *Cognitive Radio* is the next evolution in the expansion of future communication systems. The goal is to add intelligence to radio networks and provide them with additional advanced functionality. One example of that functionality, is the ability to search for available bandwidth and then transmit and receive in a new, unoccupied channel. Future applications, such as *Dynamic Spectrum Access* (DSA), would allow for radios to find and connect with

other radios to form a dynamic backbone network. This network has uses in forward operating bases (FOBs) as well as in rapid emergency response.

Signals are transmitted from essentially every powered device. In this research, active transmitters are explored for the purpose of localization. Active transmitters are devices that transmit signals using the device's own energy supply. Examples of active transmitters would include cell phones, walkie-talkies, radios, and other communication devices. Active transmitters can be detected using a variety of methods, but the simplest is energy detection. Energy detection can be performed by taking the Fourier transform of received time data to compute the Power Spectral Density(PSD). By searching the frequency data for energy spikes, it is possible to find signals that are being transmit in the environment.

Spectrum Sensing is the concept of searching through the spectrum to find which frequency bands are being used. The detection of available spectrum can be used in a variety of ways. DSA and *Cognitive Jamming* are two examples of using spectrum sensing. DSA is showing white space between primary and secondary users in both time and frequency to potentially use the available spectrum [6]. Using DSA can include more users and improve bandwidth efficiency. Cognitive Jamming is intelligently jamming channels of interest only while a source while is transmitting. This degrades the source by corrupting parts of packets to prevent reliable communication. By only focusing on parts of packets, it is possible to jam multiple frequencies with the same amount of energy [7]. A third use for spectrum sensing is finding signals of interest, such as unidentified active transmitters. Once the power of a signal is

detected, it is possible to use Received Signal Strength to localize the position of the source. This research utilizes spectrum sensing for localization within a deployable cognitive radio network, and utilizes spectrum sensing for localization.

1.3.2 Localization. Currently, the *Global Positioning System* (GPS) is the primary method of localization in both the public and private sectors. Commercially, it is used in automobile navigation systems, as well as in cellular devices. In the military, it is used in auto-pilot systems, weapons guidance, and troop tracking. While GPS has proven to be a powerful and precise method of locating users, it has also shown weaknesses. Its weaknesses are most prevalent in urban environments where walls of buildings deny Line of Sight (LOS) to the GPS satellite [8]. In these types of environments, other methods become essential to augment or replace GPS.

Another limitation of GPS is that it is used only for locating cooperative users. For example, if an individual wants to find their own position, satellite information must be collected to generate an estimate. The localization process is preformed by the user, and is not shared with others. Non-cooperative localization has applications for tracking active or passive transmitters in both the private and public sectors. Non-cooperative localization is also applicable for locating where signals are present, and then predicting where spatial white spaces exist.

There are several methods of localization that could be used when GPS is denied or when trying to locate non-cooperative targets these include: *Received Signal Strength* (RSS) localization, *Time Difference of Arrival* (TDOA) localization, and

Angle of Arrival (AOA) localization. The simplest of these types of localization is RSS localization, which is where power measurements at several nodes are used to generate an estimate of the source's location. TDOA localization uses the difference in time between a signals transmission and arrival to find the range, then estimates the location using triangulation. AOA localization relies on detecting the direction from which a signal is transmitted, with respect to the sensor. Once it determines this information, a small number of sensors can predict where the source is located [9].

This research applies distributed processing to improve the performance of the RSS localization and to improve accuracy, power consumption, bandwidth consumption, and computational time.

1.3.3 Wireless Distributed Computing and Distributed Algorithms. Sensor networks have become much more common in today's world. Networks of smart phones are a common example of sensor networks. These devices have very limited battery life and often the battery life becomes the fundamental limitation. A method of improving the energy usage is through *Wireless Distributed Computing* (WDC) [10]. By dividing a task among multiple sensor nodes, the energy usage per node can be reduced. This energy savings can then extend each sensor's battery life, and allow the sensors to stay in the field longer without recharging.

Distributed Algorithms are commonly applied to wired networks for solving complex tasks. They are fairly well understood and books have been published describing them [11]. Distributed algorithms are common algorithms that spread a task among

multiple nodes to improve processing speed and reduce the computational load on any single node. Reducing computational intensity on machines allows additional operators to more efficiently use the resources, and improve the time required for processing [11].

1.4 Research Objectives and Contributions

This section explains the contributions of this research. It also explains the products that this research generates.

1.4.1 Contributions. This research validates WDC using the example problem of RSS localization using the Maximum Likelihood Algorithm [3]. The research measures "quality of service (QoS) constraints" [1], which are tested in the form of power consumption requirements, communication requirements, such as latency and bandwidth usage, and computational requirements, such as accuracy. These are evaluated and compared via a trade space analysis. Each category is evaluated by defining the metrics to be explored. The first metric is accuracy, the second is power consumption, and the final is bandwidth and latency. Finally, initial experiments are preformed to demonstrate the feasibility of applying distributed RSS localization on a USRP2 test bed such as the CORNET at Virginia Polytechnic Institute and State University.

1.4.2 Trade Space Analysis. The trade space analysis shows the relationships between the different metrics with respect to the various the clustering tech-

niques. This analysis develops a baseline for comparing the different methods so that a system designer can select the best method for their situation. The comparisons are based on four metrics: Power Usage, Accuracy, Bandwidth Usage, and Latency. The Latency is measured from the time when a source becomes detectable until a location estimate is generated. If a user wants to favor any of these parameters, the trade space analysis presents which methods and arrangements would be the most effective. This trade space is measured and examined using a simulation.

1.4.3 Simulation Model. A simulation model is generated that allows for several different node layouts, distribution methods, and system performance metrics. For this research, a single ring topology is used. This ring is used to represent sensors placed around a fence line, or building perimeter. The simulation explores different distribution methods of grouping nodes into clusters. These clusters are adjusted to improve power performance, accuracy, bandwidth consumption, or other performance metrics. The simulation serves as the final deliverable of the research effort, and will be expandable in future research projects.

1.5 Thesis Organization

The research in this thesis is presented in five chapters. Chapter II contains a literature review of related research in the area of distributed localization. Chapter III provides the methodology for this research, which explains the experimental process that is followed. Chapter IV presents the results of this thesis, and includes all of the findings as well as a discussion of trends and interesting data points. Finally, the

conclusion in Chapter V recaps the results and provides recommendations for future work.

1.6 Conclusion

This research effort provides a potential application of WDC. Localization serves as an application of WDC, motivated by the requirement of improving both power and bandwidth usage in a deployed environment. The gains to localization will be applied to Cognitive Radio networks, because they allow for improved methods of signal spacing by using the time, frequency, and spatial domains.

II. Literature Review

This chapter presents a study of topics related to wireless distributed computing, spectrum sensing, and localization using cognitive networks. The topics described here are built upon in the remainder of the research.

2.1 Cognitive Radio and Dynamic Spectrum Access

There have been several recent advances in Cognitive Radio. New interest in this topic originates from the idea of changing the current Federal Communication Commission (FCC) spectrum usage standards to allow for more users to share the same spectrum range. The idea is that primary users would use the spectrum, and the left over "white space," between channels in both time and frequency, could be reused by secondary users. Various techniques of scanning this spectrum to monitor its usage have been devised. A large spectrum scanner atop the Illinois Institute of Technology (IIT) tower in Chicago monitors the spectrum in downtown Chicago. It generates spectrum versus time plots over a very large bandwidth. The scanner is used to determine how much of the spectrum is being used, and how often the occupied bands are being used. This project was a research effort to push for improved Dynamic Spectrum Access (DSA) [2]. While the IIT method of sensing is valid, it takes a long time for the single sensor to scan the entire spectrum. It is necessary for quick scans of the spectrum to allow secondary users to detect and utilize white spaces as they become available. One potential method of increasing the speed of the scanning process is to include more scanning nodes. A potential method is through

Cooperative Sensing, which is based on the idea of creating a sensor array whose components work together to scan segments of the spectrum at a given time [6]. The sensor array explored by Bacchus et al. used a grouping of several *Software Defined Radios* (SDR) as the sensors. Cooperative Sensing allows the single spectrum to be scanned in parallel rather than serially, saving considerably on the time and cost of spectrum sensing.

Cooperative sensing can be useful, but issues arise in network construction. Because of the large spectrum access of wireless radios, protocols must be created to allow for the radios to locate one another and create wireless links between them. To deal with this issue, Rendezvous Protocols have been developed [12]. The radios use various methods of searching the spectrum as they attempt to detect another radio's beacon signal. The beacon signal is a tone transmitted to allow for other radios to detect the transmitter. Each SDR spends some amount of time transmitting a beacon tone on an available channel, and another portion of the time listening for another radio's beacon tone. Once located, the radios can perform some simple virtual handshaking and establish a reliable method of communication [9]. Because of this reliable method of linking SDRs, it is possible to build a robust *Cognitive Radio Network* (CRN).

Approaches to optimization of DSA in CRNs have already been developed [13]. DSA is used to further develop the availability of spectrum for secondary users. One issue with optimal DSA algorithms is that they do not take into account spatial differences in the spectrum. Primary users may only transmit a signal over a few

kilometers, and outside that range the frequency is completely available to other users. DSA can be improved using localization methods to determine the spatial presence of signals.

Cognitive Radio has also been examined for its military applications. Ashwin Sampath has explored multi-channel jamming using cognitive radios [7]. Significant gains in the number of jammable channels can be achieved using only one cognitive radio. Sampath shows "that an attacker using a single cognitive radio can jam up to 7 channels. Such jamming attacks pose a serious threat" [7]. This jamming could be improved further with the knowledge of a signal's coordinate origin. It is also important to note that the current method of jamming involves filling a frequency channel with energy, raising that channel's noise floor, and corrupting the transmitted signals. To perform jamming in this fashion requires a large amount of energy. Switching to cognitive radio makes it possible to generate small collisions and corrupt enough of the source's packets to prevent reliable communication. This reduces the energy required and makes the jammer more difficult to locate.

2.2 Wireless Distributed Computing and Distributed Algorithms

This section will provide an overview of Wireless Distributed Computing and Distributed Algorithms, as well as some current research areas.

2.2.1 Wireless Distributed Computing. Wireless Distributed Computing can be applied to CRN to better leverage the network's computing potential. WDC uti-

lizes nodes that can communicate over a wireless channel to share the computational load of a complex task. Developed by Dinesh Datla, WDC utilizes the processing power of idle nodes to allow the network to perform in a more energy efficient manner. His concept relies on determining a cost function between a centralized data repository and the nodes used for distributing. This cost function is then used to determine which nodes are useful for their computational abilities based on a variety of factors [10]. Datla et al. shows that by applying WDC to a CRN, the network lifetime can be improved by over 90% in systems with four or more nodes [1]. The model used for Datla's research can be directly related to this research model. Datla's simulations apply a communication system and a computational system to monitor the energy usage at each of the nodes in the system [14]. Similar simulation methodology can be used for the distributed localization problem. Datla et al. also examines the effects of various other metrics, including latency and system scalability [10]. Their papers thoroughly explain WDC and its potential gain. They also present potential problems that must still be addressed.

The first problem that arises in WDC is the presence of additional bit errors caused by the extra wireless transmissions used in WDC. Datla et al. shows that there is a correlation between bit errors in transmission and errors that appear in the final result of the communication [14]. Bit errors are significant to note, because they must be taken into account when performing processing on the SDRs. Other problems arise when trying to implement the WDC idea on an actual platform. These problems are related to the power consumption of real world devices and synchronization problems

[1]. These problems are easily solved in simulation, but remain serious issues in real world systems and must still be addressed. While Datla et al. does explain WDC, he never applies it to a specific application or candidate algorithm.

WDC can be impacted severely by variations in channel conditions. Datla et al. continued his effort by exploring some of the effects of varying channel conditions in a WDC network. In particular, there was a reduction in performance for higher numbers of nodes. The reduction in performance occurred in both latency and power consumption for the three methods proposed. These methods included an evenly distributed workload to all nodes in the network, a tasking based on average channel conditions at the node, and a hybrid approach developed by Datla.

Comparisons must also be made to Figure 2.1 [1] to fully validate the WDC model created by Datla. It can be seen that the network lifetime improves as the number of nodes are increased, and additional energy savings are found when the back haul range is increased.

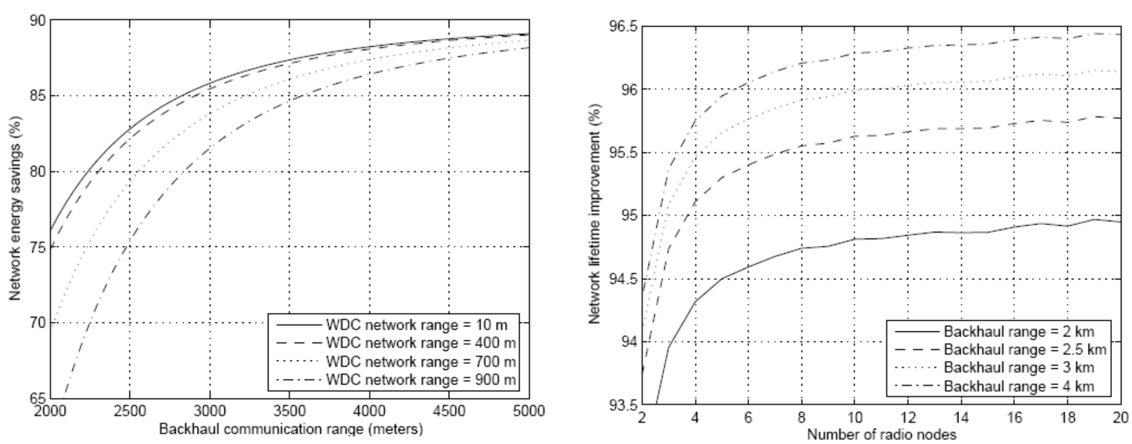

Figure 2.1: Effect on Network Energy Savings based on communication range [1].

In order to apply WDC to a system, it is important to understand the fundamentals of network connection and task allocation. Early foundation for task allocation in collaborative networks was provided by Yang Yu and Viktor K. Prasanna [15]. Their work was based on optimizing task allocations for Fast Fourier Transform (FFT) and LU factorization problems using linear programming and heuristic approaches. Their findings show that energy can be saved in collaborative networks using these methods. In some cases, they show that there are potential lifetime improvements exceeding 1000% of the original life [15]. This research was continued by Datla et al. who relates his WDC concept to Yu's and Prasanna's method using a task graph and a communication graph. The *task graph* shows how the WDC process is segmented into several software components, where each step of the distributed computing is processed. The *communication graph* shows the available connections, and the cost that each connection yields. The WDC framework developed as a cognitive engine (CE) maps the task graph to the communication graph, generating the optimum task flow for the algorithm distributed task to implement. An example of task and communication graphs are shown in Figure 2.2. Similar graphs will be developed for the RSS localization task. In the task graph, each link usually has a weight associated with it. These have been omitted here as they are represented by the required power for communication.

2.2.2 Distributed Algorithms. Distributed Algorithms can take on several different forms. Two common types of algorithm are *Wave Algorithms* and *Traversal*

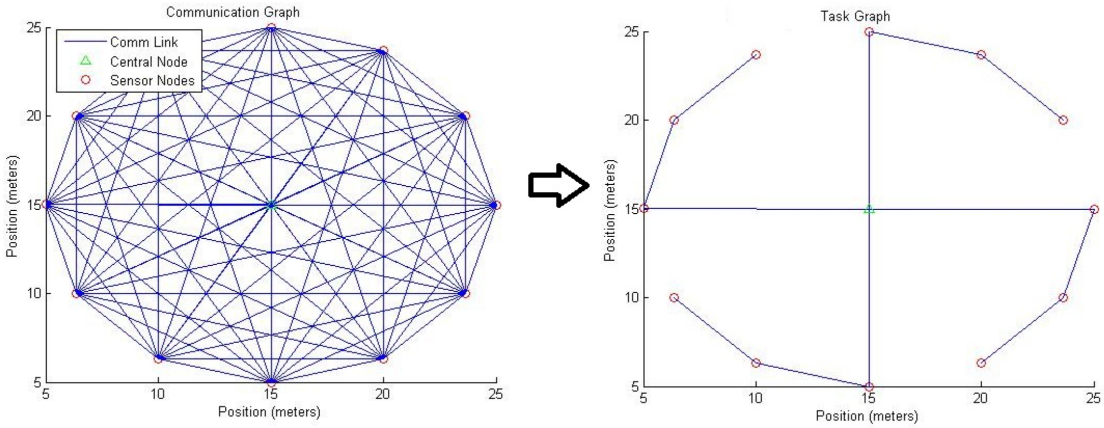

Figure 2.2: Communication Graph to Task Graph

Algorithms. A Wave Algorithm must satisfy three requirements: it must have a termination point, each computation must contain a decision, and each decision must be preceded by an event. The termination point prevents the algorithm from running indefinitely. The decision requires the algorithm to determine if enough information is available or if more nodes must be included. Finally, the algorithm must begin with an initiation event occurring. It can be used for a task that continually processes information. The *Wave Algorithm* may be centralized or decentralized: it may have one initiator or several, and it can be implemented in any topology desired. The algorithm may also require some initial information about the topology, neighbors' names, or the complexity of the system. In a Traversal Algorithm, there can only be one initiator that sends only one message; once this message is received, the next node sends out one message or makes a decision. The algorithm then terminates at the initiator node, after every node has sent at least one message [11].

Some algorithms called *Ring Algorithms*, fall into both of the categories. Ring Algorithms can be initiated by one node or multiple nodes, and nodes then send one

message to the next node in the ring. This message travels around the ring until it returns to the first node, which then decides to process the data if it has enough available environment data. It has a clear initiation and termination point, beginning and ending at the same node. The ring algorithm is an algorithm for data collection and processing, once enough data has been acquired to produce a solution to the algorithm's task. If more data is needed, then the currently available information is passed to another node in the network. The standard ring algorithm is demonstrated in Figure 2.3.

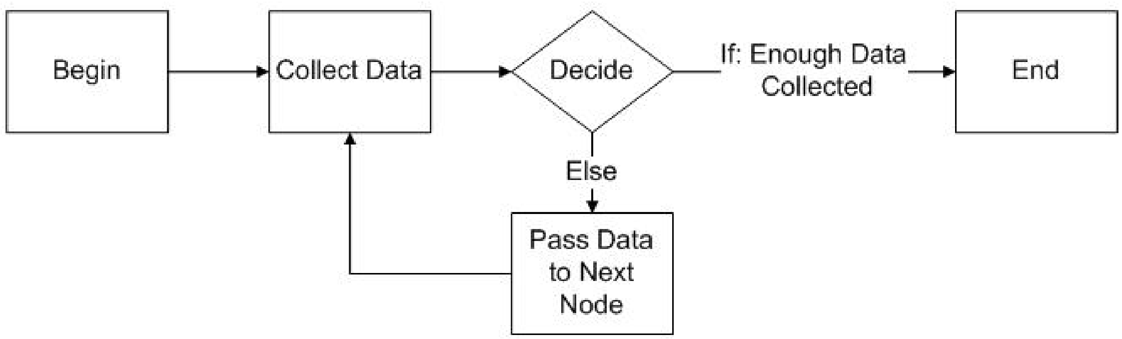

Figure 2.3: Ring Algorithm Flow Chart

In the algorithm shown in Figure 2.3, there are two types of nodes of importance. The first is the initiator, and the second is a non-initiator. The *initiator* is a node that begins the algorithm, and is represented by the begin block, sends a token or datagram to the next node. Once a decision is made the algorithm is ended. The *non-initiator* simply receives and then decides what to do. Both types of node may append additional information to the packet; in this research the additional information will be each node's RSS measurements taken from the spectrum.

Another example is algorithms that are commonly called *Tree Algorithms*, which rely on leaf nodes to pass the data towards the center of the tree. A final decision is made once the data propagates to the center of the tree [11]. Tree Algorithms are particularly useful in cases where data can be preprocessed in preparation for a final computation, and can be directly applied to the localization problem. The leaf nodes act as the initiators of the algorithm and pass the collected information towards a central node. Data is processed at each level of the tree. The algorithm for a standard Tree Algorithm is represented in Figure 2.4.

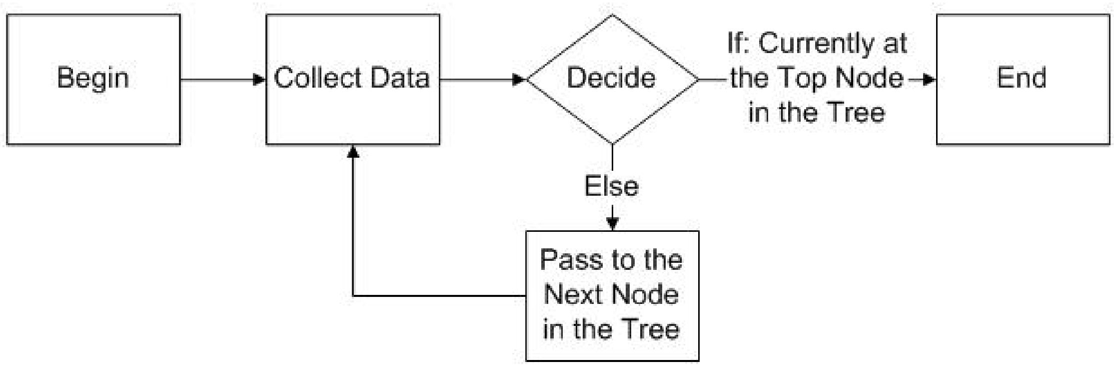

Figure 2.4: Tree Algorithm Flow Chart

In Figure 2.4 the tree algorithm is described. In the algorithm the initiator node, indicated by the begin block, waits for the initiation event. Once received, it passes data through the tree until it reaches the top level. Finally, a computation is made at the end block. Figure 2.5 shows both the ring and tree algorithm data flow diagrams.

2.2.3 Distributed Algorithms Applications. Distributed Algorithms have been applied to many problems over the years. Recent research focuses on apply-

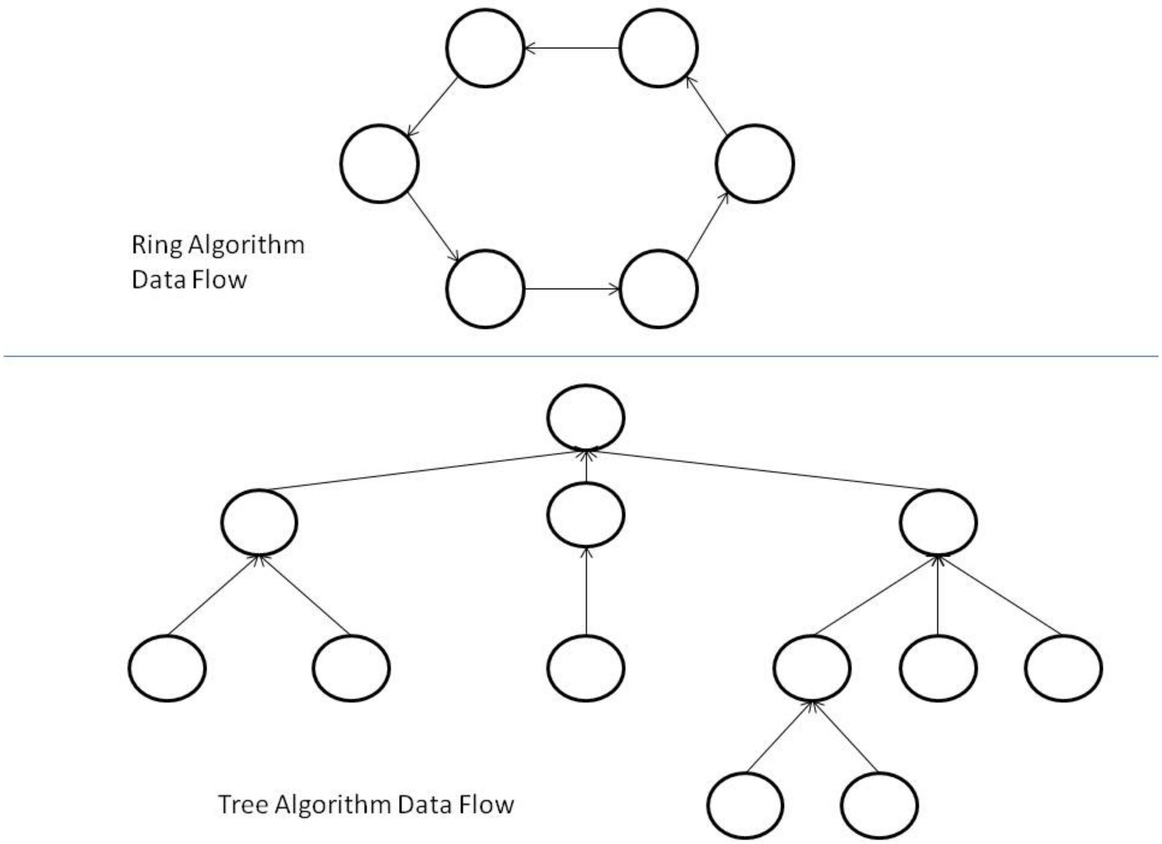

Figure 2.5: Distributed Algorithms Data Flows

ing these Distributed Algorithms toward improvement in data collection and more effective techniques for node localization.

Distributed Algorithms are often used when several processors are available to a user, and they can greatly reduce the time needed for processing, as well as reduce data storage requirements. Distributed Algorithms can also improve the power usage of the overall system, and allow for additional processing to occur. Distributed Algorithms can be used for a variety of tasks; one such example is data collection, where nearby sensor locations collect correlated data. Spreading the computation amongst multiple processors can be used to reduce the amount of data that must be

returned to a central node for processing. A one-dimensional model shows that there can be and energy conservation benefit using this correlation, and the energy used per node could be reduced by a factor of nearly six, as the number of nodes increase [16]. It is important to understand the potential energy efficiency gains that distribution provides. The correlation of data in a sensor network is critical to improving the localization problem. It also assists in determining the best clustering algorithms for generating localization estimates.

Distributed Algorithms can also be used for adaptive filtering of sensor data, to improve the quality of collected data. Using either a Least Mean Square algorithm or Recursive Least Squares Algorithm, it is possible to improve the performance of a noise canceling adaptive filter [17]. The filtering is done by sharing all information gathered by several nodes, in an attempt to better understand noise characteristics. The distributed approach was shown by Abdolee and Champagne to reach a better steady state than the centralized approaches. The filtering process may assist in energy detection of real world signals in the RSS localization problem.

Finally, localization has been explored with Distributed Algorithms for a variety of purposes. Xing-yu Pi looked at distributed target localization problems for wireless sensor networks [18]. The algorithm spread the processing out to several nodes, several estimates were created, and the centroid was then used to find the final position estimate. A block diagram explaining the algorithm is shown in Figure 2.6.

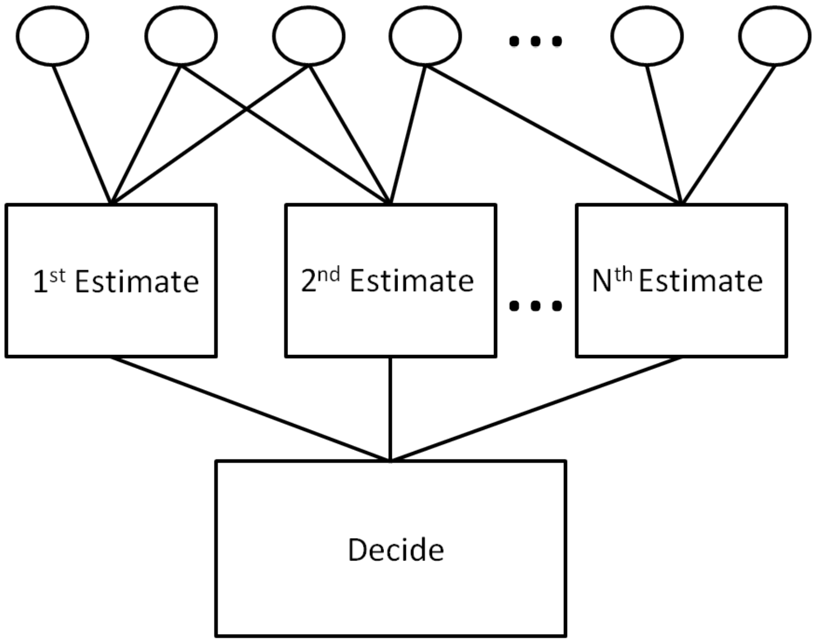

Figure 2.6: Pi's Distributed and Cooperative Target Localization Algorithm

While this method was shown to be effective, the author did not apply any fading model to communications, and assumed that accurate range measurements can be made at each node [18]. This is not true in a purely RSS method. The method of fusing the estimates using the centroid will be applied to this research. By performing multiple estimations at several different nodes, and then combining these estimates, it is possible to reduce computational intensity of the task and spread the work load to multiple nodes. Additional approaches will be further explained in the following section.

2.3 Localization Methods and Implementations

Better localization of assets has been a topic of interest in the United States Military, as intelligence about enemies is an important area of study. RSS methods of localization are an emphasis area, because of the simplicity of data collection. RSS is often used to assist in locating friendly nodes. For example, if a few nodes have GPS locations available, they can serve as anchors for other nodes in the system. Hosein Sabaghian-Bidgoli utilized the GPS anchors to improve the localization of sensors in the sensor network [19]. This improvement allowed for better localization of all nodes in a network, and improved the localization of non-cooperative sources. Using the anchor nodes, and the known communication ranges of the nodes it was possible to locate other network nodes. By knowing how far a communication signal can be transmitted, localization can be established using connectivity. If a group of nodes can connect to one another, then positions can be approximated amongst the group [19]. An example of this friendly localization is provided in Figure 2.7. In the figure, the node falls within communication range of three other nodes. The three overlapping communication circles define one central region in which the node could be located; the algorithm selects the center of this region. The left figure shows the actual position of a target. This position is slightly off-center from the intersecting region. The figure shows the prediction being centered in the intersecting region. This introduces error in the estimate. As more anchors are added, the estimate will become more accurate. While this is not using pure RSS measurements, it does require the knowledge of the transmit powers to localize the nodes. This is a valid method of

distributing the localization task; however, it can only be employed to friendly nodes trying to localize each other.

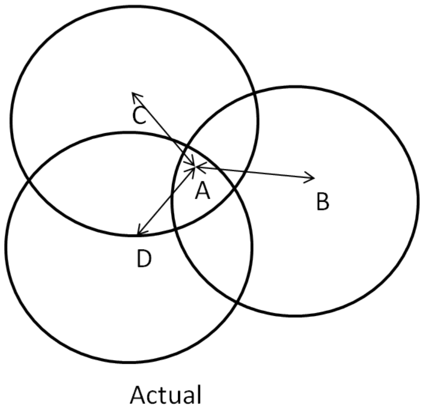

Figure 2.7: Friendly Localization Example Using Localization by Connectivity

Pure RSS measurements can be used in a number of ways to create estimates for the location of a non-cooperating node. One method uses the beacon method. First, a signal must be associated with a node at a given location. This is called *fingerprinting* the node, and establishes the fingerprinted node as a beacon [8]. Once a fingerprint database has been established containing multiple nodes, the RSS measurements of the beacon nodes can be compared to new nodes entering the *Region of Interest* (ROI). Using this information, it is possible to approximate which beacon the new signal's source is closest to, and how far away the signal is from all of the beacons [8]. The other method of applying RSS, is to collect data from a large group of sensors, and then locate the source based on RSS measurement variations from sensor position to sensor position.

It is important to recall the difference between active and passive transmitters. An *active transmitter* is any device that is intentionally transmitting a signal. A

passive transmitter is a device that is unintentionally transmitting a signal, or transmitting using energy not derived from an on-board power source. An unintentional signal is generated by most powered devices, and the power of that signal is dependent on the quality of the device design. An RFID is an example of a device that does not have an on-board power source; it begins transmitting after being activated by a RFID reader, or similar signal. Methods have been developed to locate both of these types of transmitters. Additionally signals can be used to detect solid objects, using a method called Radio Tomographic Imaging (RTI). RTI relies on data collection from a group of sensors surrounding a particular target of interest. The data collected are RSS measurements received from communication with other nodes in the network. When an object moves into the path of the communication between nodes, the RSS value will become lower. With a large collection of these measurements it is possible to create a mapping of the locations where objects are. An example is shown in Figure 2.8 [2]. In the figure, a sphere moves between a row of several sensors. Based on the sensor readings, it is possible to estimate the size of the object, and its position along the road. This is useful for detecting and locating vehicles moving down a road.

For active transmitters, an RSS value can be generated by listening to a signal of interest. By computing the power of the signal of interest, the RSS value is obtained. By combining several of these measurements, through different methods, it is possible to generate a position estimate for the signal's source of the. One particular method is using the approximate Maximum Likelihood (ML) algorithm presented by Rick Martin and Ryan Thomas [3]. This algorithm uses an estimate the location of a

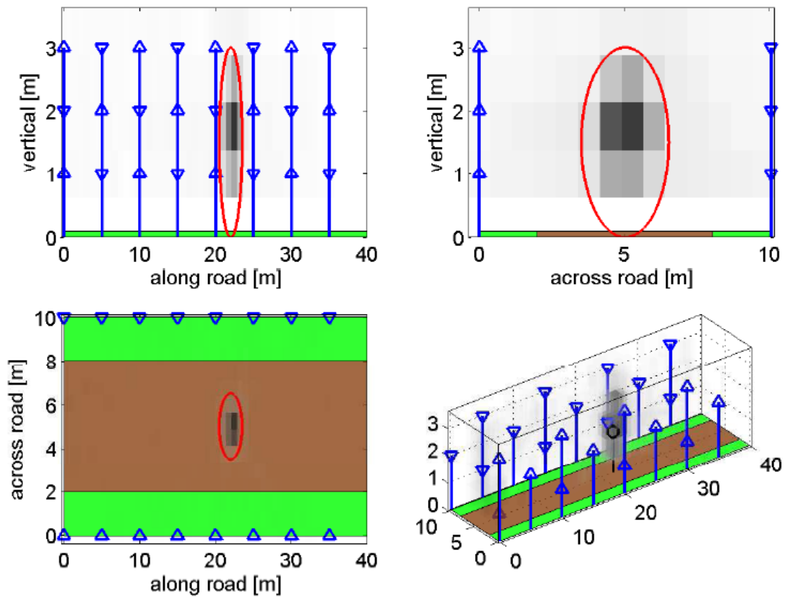

Figure 2.8: Example of RTI localization methods [2]

source, as well as its transmit power, to determine the range of communication. The algorithm can also be used to locate directionality of the source's transmit antenna; however, in this thesis, it is assumed that the transmit antenna is omnidirectional. The first step of the algorithm is to select a point in the selected search grid. Next, Equations (2.1) and (2.2) must be solved for P_0 and n_p, which represent the transmit power and the path loss exponent. These equation vary slightly from the literature, because of the assumption of an omnidirectional antenna, which eliminates two of the directionality terms [3].

$$\widehat{P_o} = \frac{\left\langle \overline{d}_s^2 \right\rangle \langle p_s \rangle - \left\langle \overline{d}_s \right\rangle \left\langle \overline{d}_s p_s \right\rangle}{\langle d_s^2 \rangle - \langle d_s \rangle^2} \tag{2.1}$$

$$\widehat{n_p} = \frac{\left\langle \overline{d}_s \right\rangle \langle p_s \rangle - \left\langle \overline{d}_s p_s \right\rangle}{\langle d_s^2 \rangle - \langle d_s \rangle^2} \tag{2.2}$$

To complete Equations (2.1) and (2.2) the variables $\overline{d}_s$ must be computed. Equations (2.3) and (2.4) describe this computation.

$$\overline{d}_s = 10log_{10}(d_s/d_o) \tag{2.3}$$

$$d_s = \sqrt{(x_s - x_0)^2 + (y_s - y_0)^2} \tag{2.4}$$

Next, the ML cost function, Equation (2.5), must be evaluated using grid location, the transmit power and the path loss parameters. The cost function must be evaluated for all coordinates in the search grid. Finally, the location estimates are selected which maximize the cost function [3].

$$L = ln[f(p|z)] = -1/2(p - m)^T C^{-1}(p - m) \tag{2.5}$$

the variable p in Equation (2.5) is a vector containing the received power from sensors in the network, and the variable m is a vector, where each element is determined by Equation (2.6) [3]. Again, this equation is varies from the literature because the directional variable, Θ, which may be omitted for the omnidirectional case.

$$m_i = P_0 - n_p\overline{d}_s \tag{2.6}$$

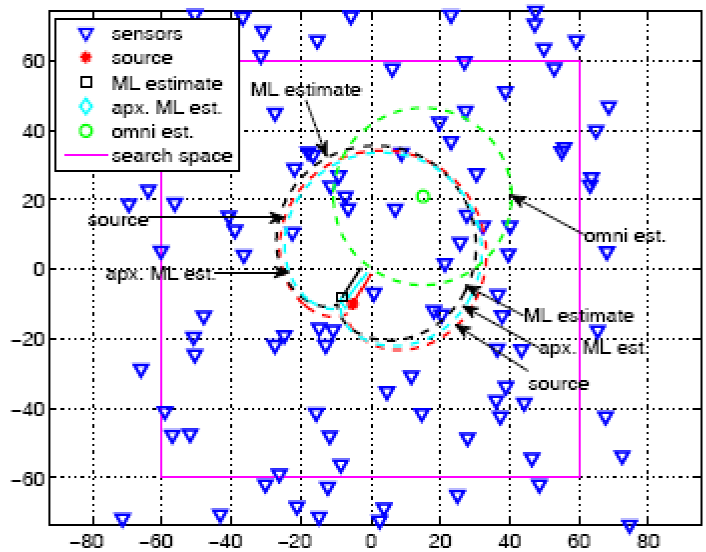

Figure 2.9: Example of Source Localization using ML Estimation Algorithm [3]

After solving the equations above, it is possible to create an accurate estimation for the source node. An example of the results collected from this algorithm is shown in Figure 2.9.

This particular algorithm is useful, because it can be easily expanded to a directional antenna, which is more likely in a real world environment. It also allows for power and path loss computations to be made, which can be used to predetermine potential interference locations on an FOB. Localizing the area of a source's communication can also be used to cognitively jam an enemy transmitter.

2.4 Conclusion

In this research, the previous topics will be combined in order to develop an improved localization system for CRNs. A CRN will be built for the purpose of localization, and Distributed Ring and Tree Algorithms will be applied to the network

for the purposes of data collection and clustering of the nodes. Once clustering is complete, localization estimates are made using the ML estimation method. By combining the preliminary distributed estimates, a final solution can be obtained. The goal is to then compare the distributed method to the current centralized ML estimation method. The results are applicable to the DSA problem, in that DSA can be performed by detecting which signals are present in what geographical locations, improving spectrum availability. Other potential uses include cognitive jamming and interference detection. Figure 2.10 provides a graphical relation for the previous work and its application to this research.

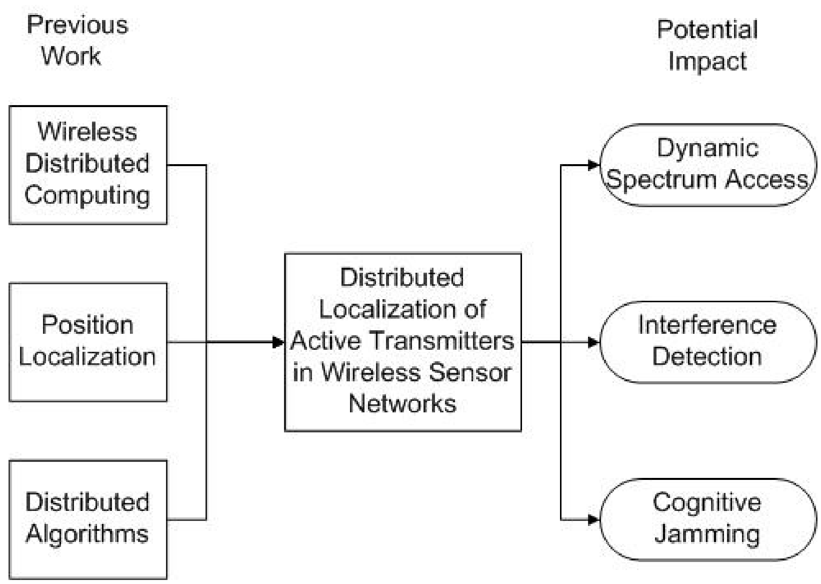

Figure 2.10: Previous Work and Application to Research

III. Methodology

This section of the thesis explains the methodology used in the research. The goal is to determine the effectiveness of distributed localization techniques and to create a comparison to centralized methods. The comparison will be accomplished according to various metrics including accuracy, power consumption, bandwidth consumption, and latency. The simulation model was validated using a real world network of USRP2 sensors from the CORNET test bed at Virginia Tech.

3.1 Simulation Model

3.1.1 Simulation Components and Subsystems. The devised simulation has several key components, which are graphically shown in Figure 3.1.

The system model will be discussed from the top of the diagram down to the lowest component. Each component exists as a MATLAB class. Each class can be instigated as an object that contains a set of properties and functions. The advantage of using this object-oriented design is that it allows for changes to be made to each component without affecting the implementation of the other classes. This implementation allows for versatility to be built into the system and enables the user to select from a vast, ever growing selection of system parameters.

3.1.1.1 Main Simulation. The main simulation controls which topologies are used, how many nodes, and when and where sources are placed. Each simulation uses a different main simulation environment to control the various parameters,

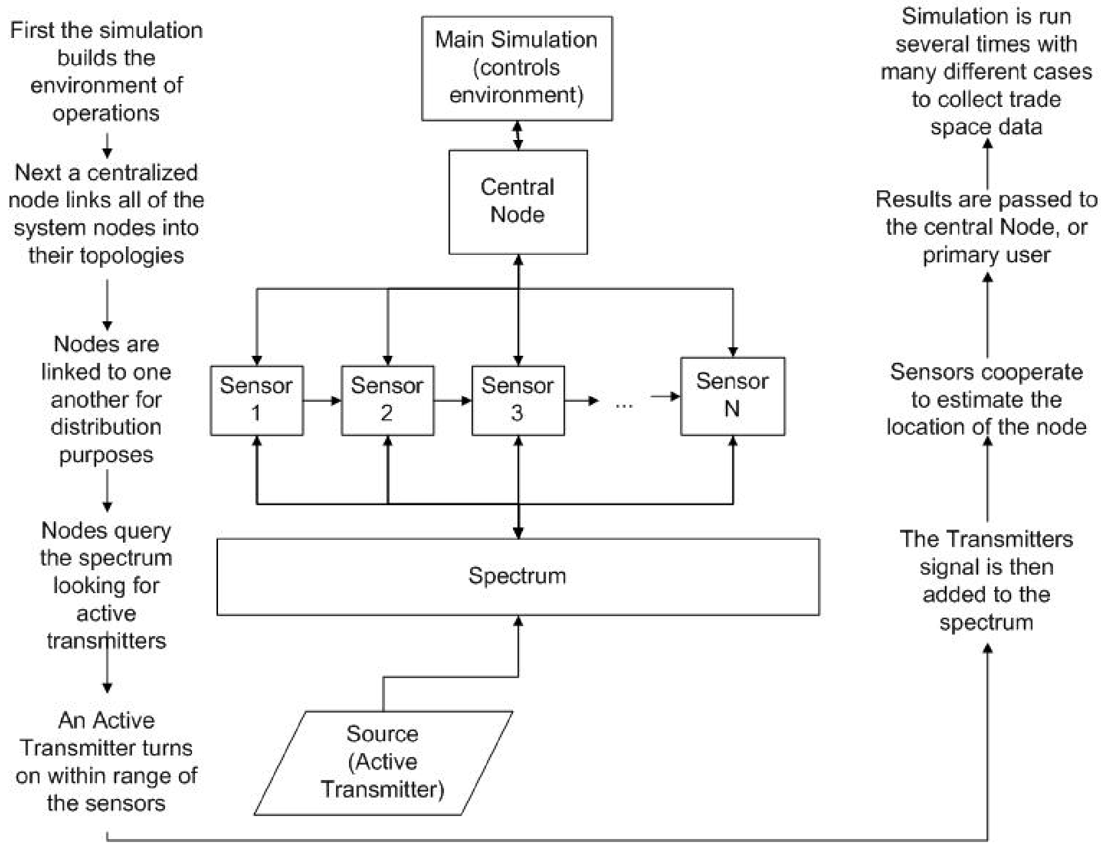

Figure 3.1: Simulation Model

and monitor the metrics of the system. The main simulation is divided into four main sections. In the first section, parameter initialization, the user enters all parameters that will be used in the simulation. These simulation parameters are shown in Table 3.1.

Additional parameters are used for determining the target's location in space. These parameters are shown in Table 3.2. Two parameters are used for checking accuracy using a Cartesian grid, and the others are for checking accuracy with polar coordinates.

Table 3.1: User Defined System Parameters

Parameter	**Range of Values**	**Description**
N	Positive Integer ≥ 12	Number of Nodes
numInit	Positive Integer $\leq N/3$	Number of Initiators
space	0,1	Allows for Selection of Next Neighbor Clustering and Node Spacing
distMode	0,1	Allows for Selection of the Estimate Fusion Method
nodeBatt	Positive Integer	Battery Life of the Sensors in Watts
sourPow	Positive Number	Source Transmit Power
nPow	Positive Number	Noise Power
BW	Positive Number	Bandwidth to be Scanned by Nodes in Hz
cFreq	Positive Number	Center frequency of the Scanned Bandwidth

Table 3.2: Transmit Location Space

Parameter	**Description**
sourXLoc	X Location of the Source, Used for Cartesian Coordinates
sourYLoc	Y Location of the Source, Used for Cartesian Coordinates
radius	Distance from Center of Search Grid, Used for Polar Coordinates
theta	Angle Between East and the Radius, Used for Polar Coordinates

These parameters are then used by simulation. Another important computation to note is the computation of the minimum number of required nodes for localization. This computation is found in Equation (3.1).

$$minToLocate = \left\lceil \frac{N}{numInit} \right\rceil \tag{3.1}$$

After generation of the critical parameters, the program moves into the next section of the main simulation, where node generation and topology are set up. The simulation creates several sensor node objects and establishes the communication links needed for operation of the sensor network. The simulation can be changed to

different topologies by adjusting the node positioning loop and establishing the source location.

The third section is the simulation section. The simulation section runs multiple trials, the number of which is established by the user defined system parameters. The simulation can either produce a grid or concentric circles. The grid approach places target locations in a rectangular pattern and transmits from each grid location. Grid source locations are useful for mapping accuracy in a graphical figure and for square topologies. The concentric circle method varies the radius and angle from the center of the map. The concentric circle source locations are useful for circular topologies, and for comparisons of different topologies according to distance from the network center.

The final section is the plot generation section, which collects data from the simulation section and produces graphical plots for the user. This section can be modified for any number of plots containing information about any of the metrics of interest.

3.1.1.2 Parent Node. The parent node has one main purpose in the simulation, which is estimation fusion. All sensors view this node as a parent, but it can be interpreted as a centralized user's computer. Each cluster of nodes generates an estimate and transmits it to the parent node. The received estimates can be fused in a number of different ways; two methods are explored in this research. A discussion of these methods is found in Section 3.2.3.

3.1.1.3 Sensor Nodes. The sensor nodes execute the largest bulk of the work in the simulation environment. Each sensor is "listening" for a signal to appear in the simulation environment. When a signal is detected, if the node is an initiator, then it begins passing data to its cluster members. If it is not an initiator, it simply waits for a datagram with information about the signal. Once the cluster of sensor nodes have gathered three or more RSS measurements, they create an estimate using an ML localization estimate, and report that information to the parent node. The sensors use a ring algorithm, previously discussed in Chapter II.

3.1.1.4 Spectrum. The spectrum contains a list of source objects that are present in the environment. When a sensor scans the spectrum at a frequency or frequency band, it reports the power in that frequency band. This scan is similar to an energy detector searching the spectrum. The spectrum object computes the effects of fading and noise on the source to each node, and is based on the source location and the node location. The fading and noise model is given in Equation (3.2); the noise is assumed to be Additive White Gaussian Noise (AWGN). In Equation (3.2), RSS is the received signal strength, γ is the fading constant, D is the distance from the source to the node, and P is the power of the source. Finally, ν is described by $\nu \sim N(\mu, \sigma^2)$

$$RSS = P - \gamma * 10log_{10}(D^2) + \nu \tag{3.2}$$

In this thesis a γ of 0.3 was used to represent an environment with a lower level of fading, because of its setup in a non-urban environment.

3.1.1.5 Source. The source object contains its location and an energy level at which to transmit. This source emits a constant energy signature at a given frequency and over a certain bandwidth. Currently, the signal is a perfect rectangle in the frequency domain, but it can be changed to improve the realism of the model.

3.1.2 Limiting Assumptions. For the simplicity of the research, a number of assumptions have been made in this research effort. These assumptions allow for a simple model to be used. These assumptions are presented here:

- Sensor nodes and target sources are both stationary and the node positions are known.
- Channel conditions are constant throughout the trial.
- Nodes have sufficient processing capability to perform the localization algorithm.
- No collisions or bit errors occur during communication.
- All nodes are identical.
- The source transmits a constant tone at a single frequency.
- The transmitter is using an omnidirectional antenna.
- All sensors are synchronized in both time and frequency, meaning there is no offset between each SDR in frequency.

These assumptions must be reconsidered when applying this to a real world environment.

3.1.3 Power Consumption Model. There are two key components to the power consumption model used in this research effort, which are modeled after Datla et al. [14]. There are two ways that energy can be used in the network. The first is through communication of data packets. Communications serve as the larger power consumption process, and are related to the distance of the communications. Equation (3.3) shows the amount of power dissipated for communication purposes, where B_{old} is current battery life, B_{new} is the new battery life, D is the distance between nodes, and P_{trans} is the power required to transmit D_o meters. In this effort, P_{trans} is set to 32mW, because of its similarity to WiFi transmission power for a 20 meter communication region, with D_o set to 10 meters.

$$B_{new} = B_{old} - 10^{\left(\frac{D}{D_o * log_{10}}(P_{trans})\right)} \tag{3.3}$$

The other energy usage comes from the processing and idle time of the sensor. The energy is computed by reducing battery power every time the sensor is used based on how long it is active. The power consumption model assumes that the processor usage is roughly the same whenever the sensor is running. Additional power is used when a localization estimate is made and must also be taken into account. There is additional power consumption when the processor performs the localization computation. Equation (3.4) shows the power consumption based on time operated, and Equation (3.5) shows the power used for localization estimates. These three calculations are subtracted from the available battery of the sensors, and can be used

to calculate the power usage of the system. Here B_{old} is the current battery life, B_{new} is the new battery life, P_{idle} is the power used while idle and is varied between two settings, and P_{comp} is the power required for computation and varied between 2 different settings. The first setting is with P_{idle} set to 1 mW and P_{comp} to 5 mW, which yields a low *Computation to Communication Ratio* (CCR). The CCR represents a comparison of the computational intensity. Then P_{idle} is raised to 5 mW for the middle and high CCR levels. P_{comp} is set to 30 mW and 50 mW for these settings respectively.

$$B = B - P_{idle} \tag{3.4}$$

$$B = B - P_{comp} \tag{3.5}$$

3.2 Topologies, Clustering, and Estimate Fusion

The topologies used and how nodes are clustered are the fundamental difference between each of the simulations in this research effort. Two topologies will be explored, Ring Topologies and Grid Topologies. The nodes will also be clustered in two different ways, Next Neighbor Clustering and Node Spacing Clustering. The distributed method produces multiple estimates; methods of fusing these estimates will also be discussed.

3.2.1 Topology. The Ring Topology is based on the idea of grouping sensors placed in a large circle, similar to a fence line around a base. This topology will serve as the preliminary experiment for verification of the model, as well as some baseline ideas of how distributed techniques will compare to the centralized methods. The Ring Topology features N nodes which are equally spaced from each other, and have a constant radius from the center of the circle. An example of a ring topology is shown in Figure 3.2.

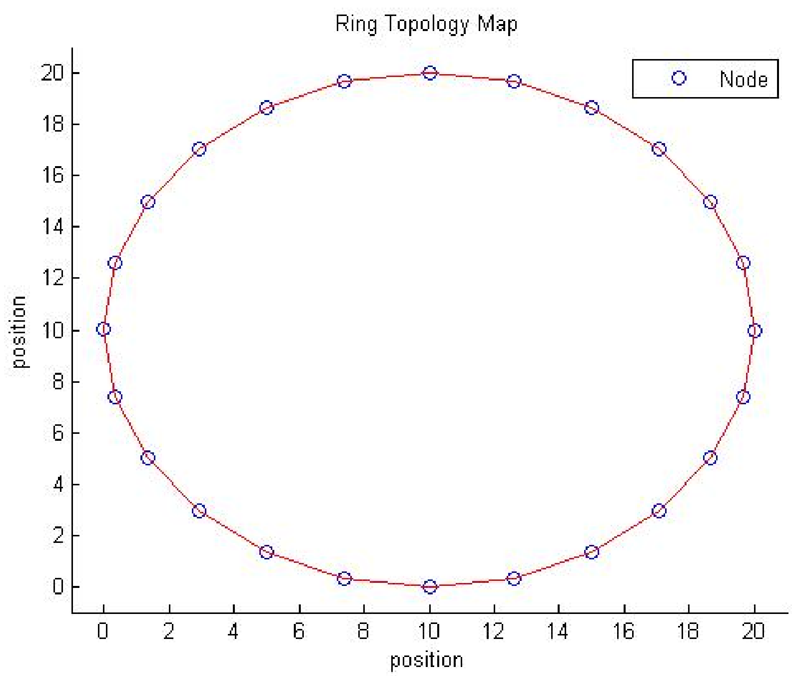

Figure 3.2: Ring Topology with 24 Nodes

3.2.2 Clustering. Two methods of clustering have been selected for this research effort. The goal of the first method is to optimize power consumption in the system. The others is to optimize the "usefulness" of the collected data by reducing the similarity of collected RSS measurements. This can be done by reducing

the number of similar sensor locations. In WDC, the ultimate goal is to optimize power efficiency, but it may be necessary to alter this goal to also achieve the computational requirements of the problem. When discussing Clustering it is important to distinguish the two types of nodes used, *Forwarders* and *Estimators*. A Forwarder is a node that takes an RSS estimate and then forwards, or passes that information to the next node in its cluster. The Estimator takes an RSS estimate, receives all of the other RSS estimates, and then performs the ML Estimation algorithm on the data to generate a localization estimate.

3.2.2.1 Next Neighbor Clustering. The Next Neighbor Clustering method is used to reduce the transmission distance between the nodes in the cluster. Nodes are grouped based on their proximity to one another. This reduction in distance will reduce the energy footprint of the system of nodes; however, it means that each cluster will collect more similar data. This correlation may negatively impact the accuracy. This method of clustering is shown in Figure 3.3. Each shape and color identifies a different cluster. Each of these nodes in a cluster collects an RSS measurement. The data is pooled and an estimate is generated from each cluster. Each result is then returned to the central node for fusion.

3.2.2.2 Node Spacing Clustering. The Node Spacing Clustering method attempts to minimize the correlation among measurements taken in the system. To minimize the similarity in measurements from similar locations, nodes that are clustered together are spread out over the ring or grid. The increase in communication

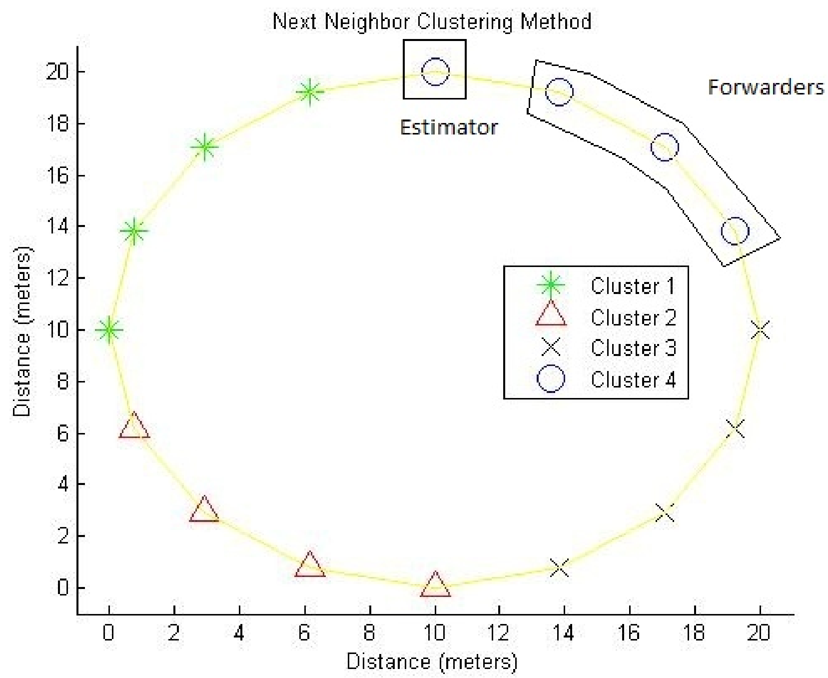

Figure 3.3: Next Neighbor Clustering Example

distance will reduce energy efficiency; however, it improves the accuracy over the Next Neighbor Clustering method. This clustering method is shown in Figure 3.4. Each shape and color identifies a different cluster.

3.2.3 Estimate Fusion Techniques. The number of initiators in the system determines the number of estimates generated in the network. These estimates must be fused to create meaningful results. Two methods have been selected for the purposes of this research. This fusion will take place as the final computation in the system task graph. The two methods explored are the Centroid method and the Weighted Averaging method.

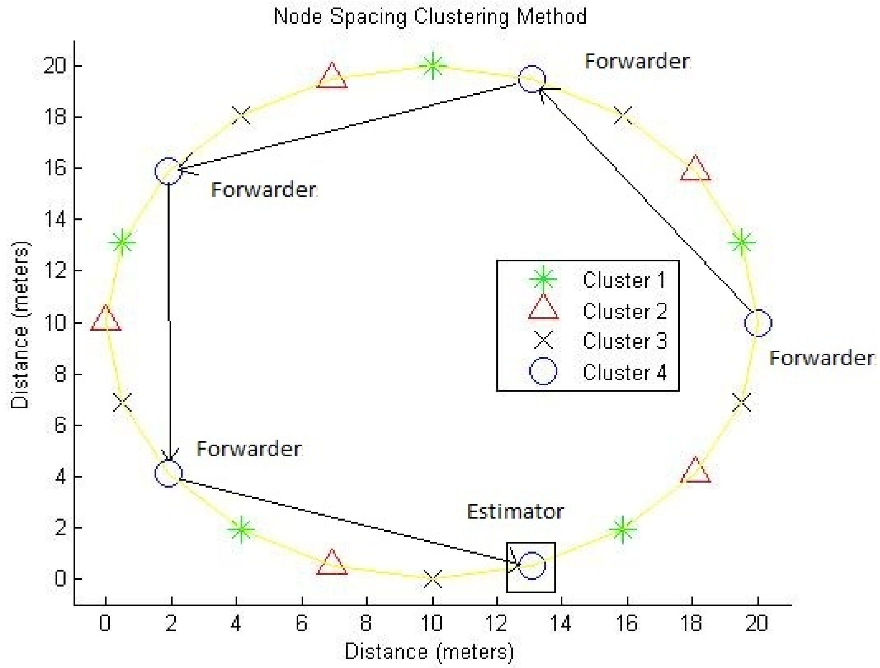

Figure 3.4: Node Spacing Clustering Example

3.2.3.1 Centroid. The Centroid method of Estimate Fusion is the simplest explored in this research. The value n represents the number of clusters in the system. The n estimates that are generated from the n clusters are divided into X and Y coordinates. The X coordinates are then averaged, as well as the Y coordinates. This result creates the final estimate. Equation (3.6) demonstrates how the estimates are mathematically fused, where $\overline{X}$ and $\overline{Y}$ are the X and Y components of the final fused estimate and X_i and Y_i are the individual component of the i-th estimate.

$$\begin{aligned} \overline{X} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} X_i \\ \overline{Y} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} Y_i \end{aligned} \tag{3.6}$$

An example of the centroid estimate fusion technique is shown in Figure 3.5.

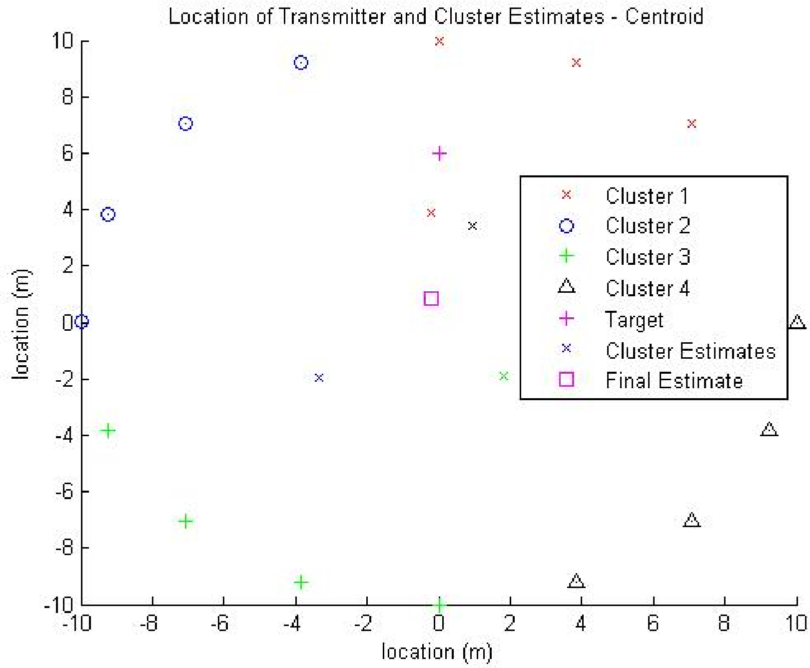

Figure 3.5: Centroid Method of Estimate Fusion

3.2.3.2 Weighted Averaging. The idea behind Weighted Averaging is that the closer a cluster is to the source, the more accurate the localization is, while the further away the cluster is, the more likely errors exist. This accuracy variance increases as the received SNR becomes lower. A cluster that is close to a source approximates the source as very close to itself, while a distant source approximates it as further away. Using Weighted Averaging allows the system to predict which cluster is more likely to be correct. How the weighting is performed is shown in Equation (3.7) where $\overline{X}$ and $\overline{Y}$ are the X and Y components of the final fused estimate and X_i and Y_i are the individual component of the i-th estimate.

$$\overline{X} = \frac{\sum_{i=1}^{n} w_i X_i}{n \sum_{i=1}^{n} w_i}$$
$$\overline{Y} = \frac{\sum_{i=1}^{n} w_i Y_i}{n \sum_{i=1}^{n} w_i} \tag{3.7}$$

An example of Weighted Averaging Estimate Fusion Technique is shown in Figure 3.6.

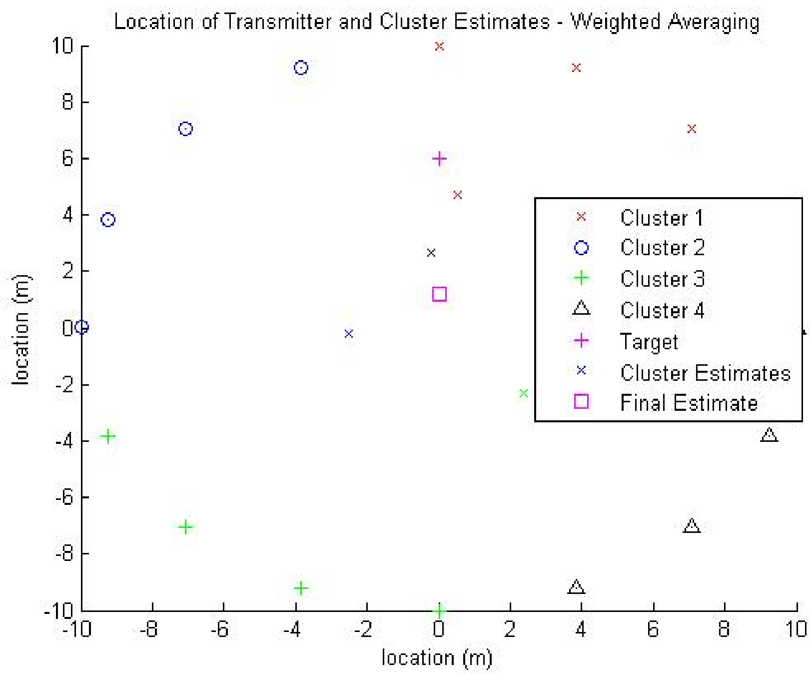

Figure 3.6: Weighted Averaging Method of Estimate Fusion

Equation (3.7) is solved using a weighting function. The weighting is computed using Equation (3.8).

$$w_i = (100 - 100 * d_i/(\sum_{k=1}^{n} d_k))^2 \tag{3.8}$$

The weighting function shown in Equation (3.8) was created to more heavily weight estimates that are found to be closer to one cluster than another. This would

likely show that the cluster has a lower signal to noise ratio received from the transmitter, and more likely to be correct. The variable d_i represents the distance from the cluster to the estimate. The variable d_k is the distance from any cluster to that same estimate.

3.3 Performance Metrics

Several performance metrics are used to analyze the trade space developed, when comparing centralized RSS localization methods with various methods of distributing the task using WDC. These metrics have been selected to validate the QoS requirements that were provided as critical to WDC. Those metrics are reviewed, and their computation is explained. Table 3.3 gives a brief synopsis of these metrics and the requirements that they satisfy.

Table 3.3: Performance Metrics Synopsis

QOS Requirement	**Metric Name**	**Units**	**Description**
Computational	Accuracy	Meters	Average euclidean distance between target and estimation
Power/Energy	Power Usage	Milliwatts	Power usage per node total and system Power Usage
Communication	Channel Usage	Channels	Channels required for a given system latency
Communication	System Latency	Time Steps	Time required for localization given a bandwidth constraint

3.3.1 Computational Requirements - Accuracy. The accuracy of the system is an important attribute to measure and understand. To determine the accuracy of

a system, the source must be moved to a large number of locations, and each area must be tested with varying levels of source power levels. The *Euclidean Distance Error* (EDE) can then be computed based on the euclidean distance between the localization estimate and the source location. Several iterations of this process must be completed to provide a 95% confidence interval. The accuracy metric provides insight into the impact of applying WDC principals to localization.

3.3.2 Power Requirements - Power Usage. Energy Usage monitors how much total energy is consumed throughout the entire localization process. This metric can be compared using three different methods. The first is by exploring the total power consumption of the system. The total power metric is ideal for comparisons that require low total energy footprints, particularly in regions with limited power availability. The second method is comparing the maximum and minimum energy used at any particular node. Some nodes may be required to use more energy than other nodes, and may serve as a limiting factor in some systems. Finally, power can be compared as the Power Usage per Node, which averages energy consumption across all nodes in the network. Average power per node is useful for networks of homogeneous nodes, like small sensors scattered throughout a field. The average power per node is used to validate the WDC concepts and how it impacts system power consumption. Comparisons are drawn relating these results to the results of the WDC system.

3.3.3 Communication Requirements - Bandwidth Usage. This metric explores how much bandwidth is required to operate the system. Bandwidth usage

varies based on the bandwidth available to the CRN. It is assumed that nodes can find and build new connections in the available spectrum. This metric is measured using a number of communication channels, and is related with the System Latency metric. This metric monitors the number of channels or links that are used. Each channel or link is represented as a dedicated section of bandwidth for the sensors to communicate over. Exact bandwidth values are determined in Chapter IV.

3.3.4 Communication Requirements - System Latency. For various bandwidth availabilities, different numbers of channels can be located. If another bandwidth is available, or nodes are spatially separated enough, it may be possible for multiple nodes to communicate simultaneously. System Latency measures the number of time slots that are required to perform the given localization problem. Each time slot represents a transmission or a location estimation. The assumption is that the system is operating in a synchronous fashion. This metric, along with bandwidth usage, provides insight into the potential improvements in bandwidth efficiency and latency.

3.4 Simulations

Simulations were run to test the metrics previously discussed. Each simulation served the purpose of comparing the metrics for the various methods of clustering and topologies. Each of these simulation's results are also be compared to the current centralized RSS techniques.

3.4.1 Simulation Cases. Several simulations were explored to explore the various metrics obtained for both centralized and decentralized techniques. To ensure consistency, each simulation was run using identical parameters, and only the topology or distribution method was changed. The Bandwidth and Latency metrics are also validated using basic networking theory. The experiment's run for each metric are shown in the corresponding section.

3.4.1.1 Accuracy Simulation Cases. Accuracy was tested in a number of different ways. First, it is important to understand the effects on accuracy in different regions of the localization map. It is possible that some techniques perform better in terms of accuracy closer to the edge of the map, or closer to the center. A comparison of how each method performs, based on the location of a transmitting source, can be valuable for a system designer. This analysis was performed by moving the target away from the center of the map at varying angles. The distance and angle are delineated as r and θ. Figure 3.7 shows this approach for comparison.

The first set of experiments test how different regions perform and compare those regions to one another. Ninety-five percent confidence intervals are also established to assist in determining the regions of performance. The different methods of estimate fusion are also compared. The experiments are described in Table 3.5.

The experiments in Table 3.5 will be run with 12, 16, 20, 24, 28, 32, 36, 40, 44, and 48 nodes with a search space of 40x40 meters. Each simulation is run with 4 initiators and each cluster contains between 3 and 12 nodes. Additionally, the radii

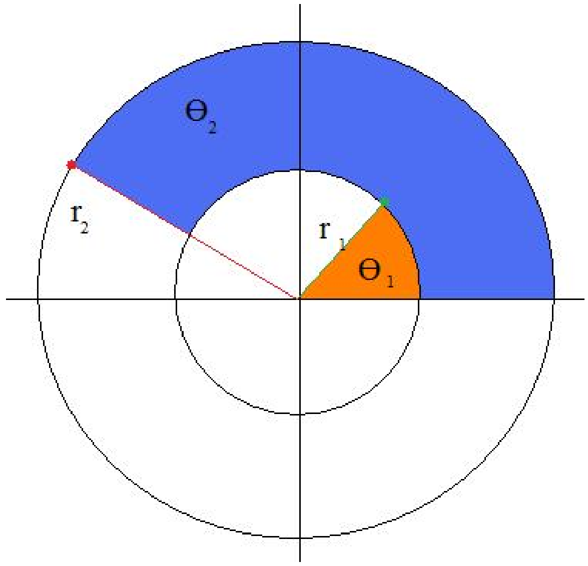

Figure 3.7: Radial Analysis Concept for Comparison of Methods

Table 3.4: Accuracy Simulations for Regional Analysis

Trial Number	**Topology**	**Clustering Mode**	**Estimate Fusion**	**Nodes**	**Clusters**
1	Ring	Node Spacing	Centroid	12-48	4
2	Ring	Node Spacing	Weighted Average	12-48	4
3	Ring	Next Neighbor	Centroid	12-48	4
4	Ring	Next Neighbor	Weighted Average	12-48	4
5	Ring	Centralized	N/A	12-48	N/A

from the center of the ring to the source were varied from 0 to 20 meters in half-meter increments. The angle will also be varied from 0 to 2π radians over four different angles, 0, $\pi/2$, π, and $3\pi/4$. Using this information, Figure 3.8 can be generated which shows the search space, sensor locations and transmitter locations on one map. In Figure 3.8, each red triangle falls on one of the black rings, which are spaced by a 0.5 meter difference in radius.

Position estimates from each of those angles are tested for statistical independence. To determine independence, confidence intervals were placed around each

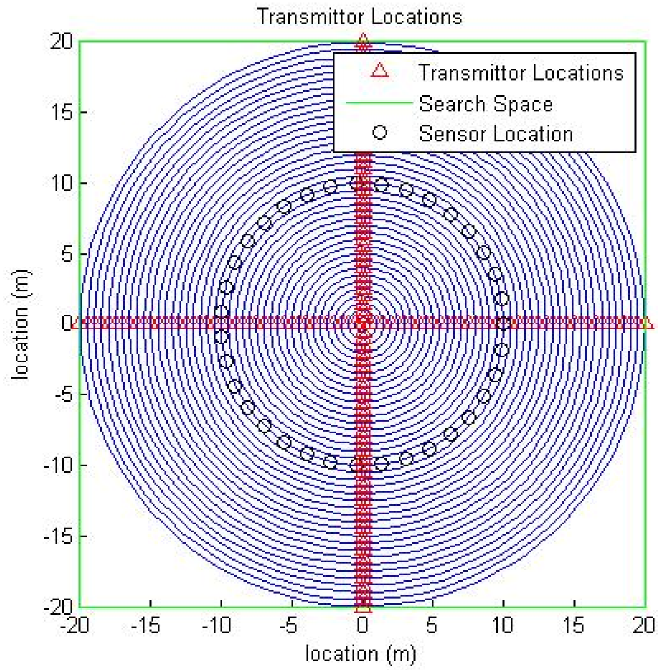

Figure 3.8: Transmitter Locations

estimate at each θ. If they are statistically the same, the results are independent of θ and can be averaged. The evidence for identical distributions on θ is shown in Appendix A. The 95% confidence intervals are used and computed according to Equation (3.9) [20], where $\overline{x}$ is the average of the trial, n is the number of trials, and $z_{\alpha/2}$ = 1.96 for 95% confidence intervals. One hundred trials has been selected, because it yields confidence intervals of less than one meter for a majority of the predicted trial cases.

$$\overline{x'} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \overline{x} \leq \overline{x'} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \tag{3.9}$$

After the accuracy has been determined and the regional performance established, it is possible to determine which method performed best. This can be done

using a number of methods. One method that can be used to determine an overall best performance, is to compute the average error of each method at each location. This method is shown in Equation (3.10), where r is the maximum radius used for transmitter locations, and in this case is 20 meters. By comparing each of these values, the average improvement of performance over the centralized method can be generated.

$$\frac{\sum_{p=0}^{r} ErrorDistance(p)}{r+1} \tag{3.10}$$

The best method, however, may be different based on what region is being considered, as well as other factors. The number of initiators must be varied to determine the impact of more clusters on the accuracy. To test the effect of varying the number of initiators, which represents the number of clusters in the network, a ring network was be generated with 36 nodes. The accuracy was then measured at all radii from 0 to 20 meters with 0.5 meter increments. The different clusters' plots will be overlayed to allow for analysis of the effects of varying the number of clusters. Table 3.5 shows the proposed experiments.

Table 3.5: Accuracy Simulations for Testing Effects of Number of Clusters

Trial Number	**Topology**	**Clustering Mode**	**Number of Clusters**
6	Ring	Node Spacing	3-12
7	Ring	Next Neighbor	3-12

The final experiment verifies which regions perform better in each clustering method. A single radius source location is used with a varying number of nodes. This

allows for a more precise understanding of the effects of increasing the number of nodes in the network at important points for accuracy. The points selected are at one and ten meters. The one meter mark represents the central area of the network. The ten meter mark represents a source transmitting at the fence line. Table 3.6 shows the experiments used for determining the effects of network size on accuracy on search area points.

Table 3.6: Accuracy at a Single Transmitter Location

Trial Number	**Clustering Mode**	**Transmitter Radius (m)**
8	Node Spacing	1
9	Node Spacing	10
10	Next Neighbor	1
11	Next Neighbor	10
12	Centralized	1
13	Centralized	10

The SNR is determined using SNR $= \frac{Ptrans}{\sigma^2}$ and is held constant at five. Also, the number of initiators can never exceed $\frac{1}{3}N$ where N is the number of sensors in the topology.

3.4.1.2 Power Simulation Cases. Power Consumption is highly dependent on the number of nodes in the system, the size of the ring, and the method of clustering. each of these variables are changed to determine the power efficiency of the system. To compare the power consumption of different systems, several simulation trials are compared based on the power usage for a specific topology and clustering method, while varying the number of nodes. Table 3.7 shows a list of test cases explored. Each of these figures compare the power usage to the number of nodes.

Table 3.7: Power Simulation Test Cases

Trial Number	**Topology**	**Clustering Method**	**Number of Nodes**
14	Ring	Node Spacing	12-1000
15	Ring	Next Neighbor	12-1000
16	Ring	Centralized	12-1000

Each of the experiments explained in Table 3.7 was run using a 40m x 40m search grid, with a ring radius of 10 meters, and a resolution of 1 meter. This experiment allows a comparison to be made for how the number of nodes in the network affects the power usage, and gives a baseline for comparing each clustering method to one another.

Additionally, a similar experiment is run to explore the effects of changing the number of clusters in the system. This is done by holding the number of nodes constant at 36 and then varying the number of clusters from three to twelve. This allows for power results of networks with different numbers of clusters and cluster sizes. For both experiments the minimum, maximum and average power consumed per node are recorded. The experiments to determine the effects of the number of clusters on power consumption are shown in Table 3.8.

Table 3.8: Power Simulation Test Cases

Trial Number	**Topology**	**Clustering Method**	**Number of Clusters**
17	Ring	Node Spacing	3-12
18	Ring	Next Neighbor	3-12
19	Ring	Centralized	3-12

3.4.1.3 Bandwidth and Latency Simulation Cases. Bandwidth and Latency are directly related to one another. As the amount of available bandwidth increases, the latency of the system decreases, and the opposite trend is also true.

Each method of clustering and topology has different bandwidth usage properties based on the required communication range technique. A mathematical analysis of what the usage is presented in Appendix B. The derived equations are then be applied to the simulation model, to provide an understanding of the bandwidth and time relationships of the distributed methods compared to the centralized method. Plots are generated showing a comparison between the number of nodes, the number of clusters, and the number of available channels. One channel is described as a link between two nodes. This link allows for a stream of packets to be transmitted reliably, and within an allotted time slot. The time slot is long enough to allow for either a scan and communication of packet data on the link, or a scan and a computation of collected data to provide an estimate. Table 3.9 shows the parameters used to determine the number of time steps required. The number of time steps represents the system latency.

Table 3.9: Bandwidth Latency Simulation Data

Trial Number	**Clustering Method**	**Number of Clusters**	**Available Links**
20	Node Spacing	1-12	1-30
21	Next Neighbor	1-12	1-30
22	Centralized	1-12	1-30

3.4.2 Trade Space Analysis. The trade space analysis comparing the centralized and distributed localization techniques requires data collected for the various simulation cases described above. Analysis of optimal conditions for each clustering method is made as well as cross comparison of each method. The final results are tabulated and analyzed for improvements in each of the various metrics. Finally,

recommendations are made as to which method performed the best and according to which metrics.

3.5 Concluding Remarks

This research effort was performed using a three step approach. First a simulation model was built according to the model shown in Figure 3.1. Next, this model was used to better understand the effects on four key metrics—accuracy, power usage, bandwidth consumption, and time to localization. Finally, the simulation model was validated using a real world testbed. The use of 95% confidence intervals allowed for easy comparison of each simulation case, and gave an accurate trade space analysis of the different localization methods.

IV. Results

This section of the thesis provides an overview of the analysis and results, and concludes with a summary of the key findings. The research areas explored are the three QoS metrics described by Datla. These categories are Computational, Power Consumption, and Latency.

4.1 Communication and Task Graphs

In order to apply WDC to the localization problem, it is important to identify both the communication graph and the task graph for the network. The following communication graph is assumed for all of the networks involved. It is a fully interconnected communication graph, and shown here in Figure 4.1.

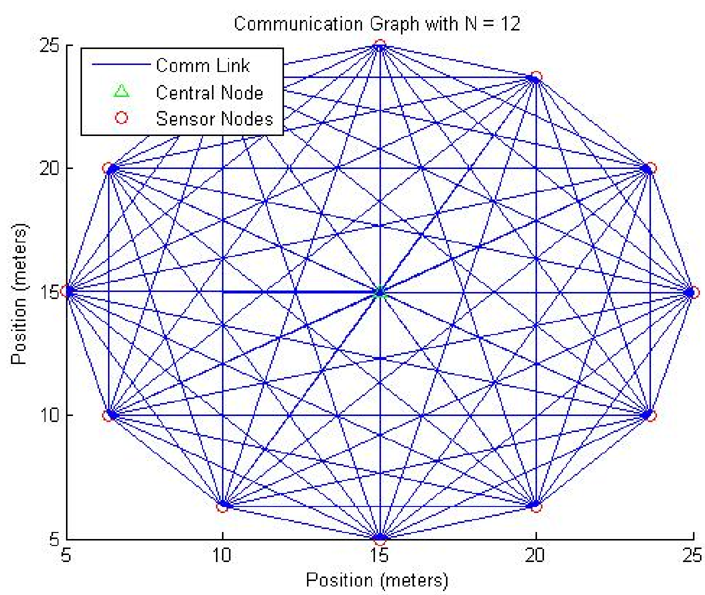

Figure 4.1: Assumed Communication Graph for all Sensor Networks

This fully connected communication allows for any number of clustering methods to be developed. The first task graph that was used is the task graph correspond-

ing to the Next Neighbor method. This is the optimal task graph in terms of energy efficiency, because it seeks to minimize the communication distance between sensors. This graph is shown in Figure 4.2.

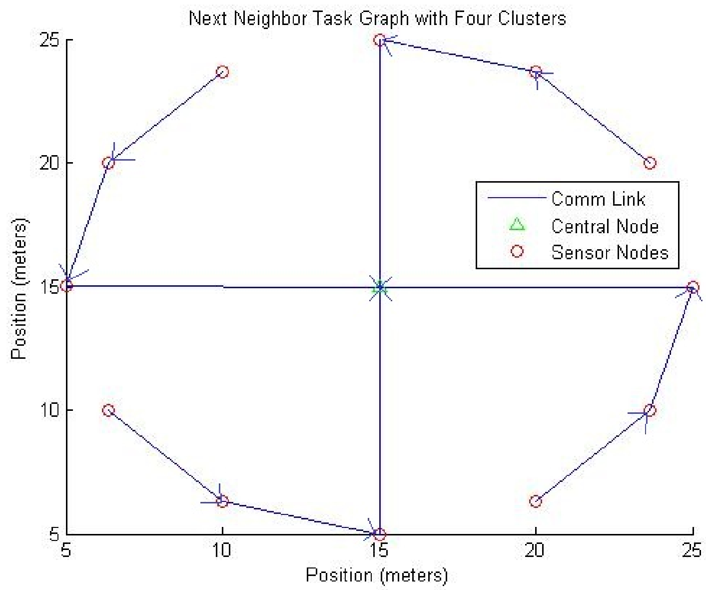

Figure 4.2: Next Neighbor Task Graph

This task graph has four clusters and twelve nodes. Each of the clusters are connected with a communication link. These links are based on a cognitive network, and they may change as channel conditions in the radio environment also change. The cost of using each communication link is based on the energy required to transmit, and is related to the distance between nodes in the network. The other clustering method explored is shown in Figure 4.3.

This task graph values the data collection process more than communication efficiency. Similar to the previous graph, each node in a cluster is connected with a communication link. Additional spacing between nodes is used to reduce correlation

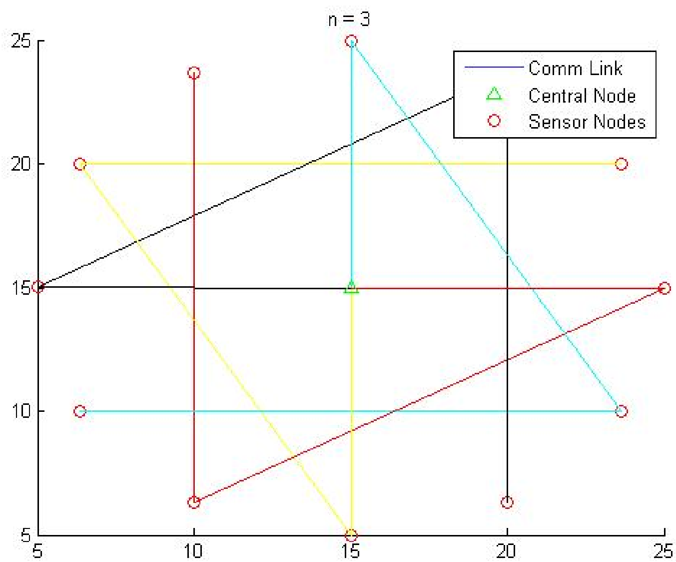

Figure 4.3: Node Spacing Task Graph

in collected data. This will likely improve the overall results in terms of computational effectiveness, but it will also reduce the performance with respect to power consumption.

4.2 Computational Requirements - Accuracy Results

In determining which system performs the best for a given situation, the accuracy results are critical. The accuracy curves show how well a system performs based on the distance from the center of the network. Depending on the application, one system may be more or less effective. For example, a system that is highly accurate near the edge of a sensor network, would make it more effective for base defense and security. Another system, that is more accurate near the center of a sensor network, may be better used for interference prevention or asset tracking. The accuracy results are replicated here for each of the methods explored in this thesis. After each method

is reviewed, a comparison is given and explained in the final part of this section. Finally, a description of the most effective method is given according to the the accuracy metric.

4.2.1 Centralized. In the centralized case, all collected RSS measurements are transmitted wirelessly to the central processing node. This simulation took the average of 100 trials with a radius parameter of 0 to 20 meters in 0.5 meter increments. The angle varied from 0 to 2π radians in increments of $\frac{\pi}{2}$ radians. This was repeated for several sensor network sizes, ranging from 12 to 48 nodes in increments of four nodes. Figure 4.4 shows the obtained results for the centralized method.

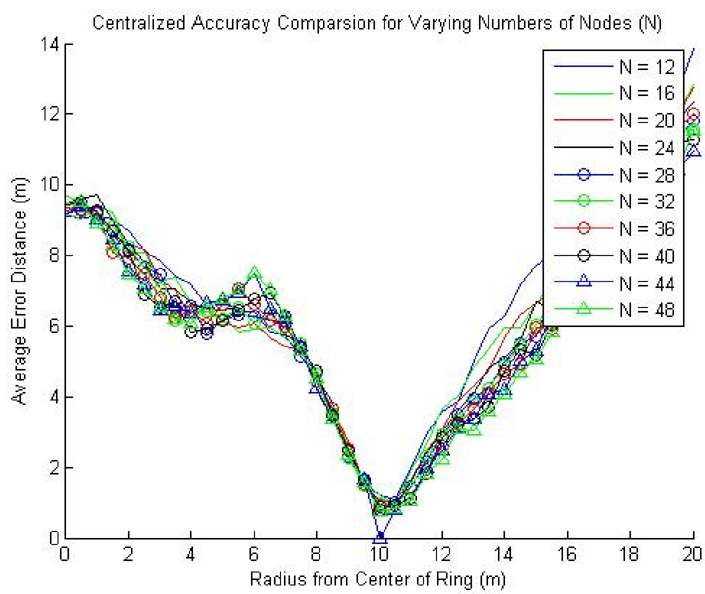

Figure 4.4: **Trial 5** - Baseline Centralized Method of Localization

In Figure 4.4, there are no noticeable improvements from one experiment to another except outside the sensor ring. This means that performance remains consistent regardless of the number of sensors used in the centralized method, and is likely

caused by the symmetry of the problem. This is important to note, because it does not fully adhere to current theory. This is further tested in the analysis section, and proven that it does follow theoretical trends. The optimum accuracy point is at the radius of the ring, which means that the centralized method operates the best the closer the target is to the sensor network. This could be useful for perimeter defense.

4.2.2 Next Neighbor. Two methods of Estimate Fusion were applied to the collected data. The first was the Centroid Method, which simply averaged the collected data points together. The results for various N's are shown in Figure 4.5. In this figure, the ring forms a 10m circle around the center, radius zero. A radius of 10m indicates the edge of the ring, while radius zero indicates the center of the search space.

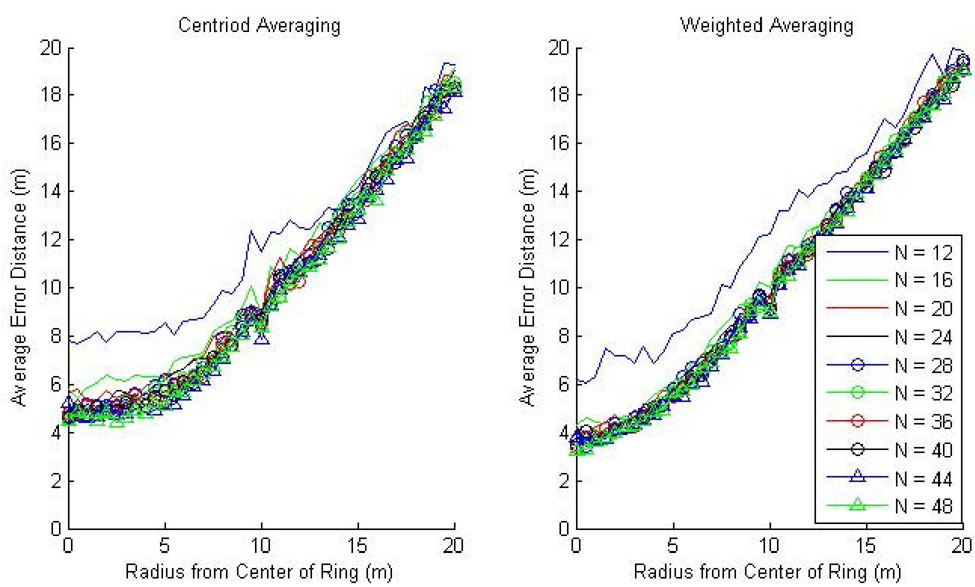

Figure 4.5: **Trials 3 and 4** - Next Neighbor Localization Using Centroid and Weighted Averaging with 4 Clusters

A similar result to what was found in the centralized results for the Centroid Averaging method, shown on the left of Figure 4.5. Here, the performance is independent of the number of nodes in the network. Also, Next Neighbor clustering performs better near the center of the ring, and then the accuracy becomes worse as the transmitter moves radially away from the center. The second method of Estimate Fusion was the Weighted Averaging Method, shown on the right of Figure 4.5. This method weights estimates that are closer to a cluster higher than the estimates that are further from the cluster. There is no significant change between each curve for the Next Neighbor case. There is a slight improvement in the performance for a transmitter location radius of less than 10 meters. This was found to not be statistically significant according to 95% confidence intervals.

By comparing the two plots shown in Figure 4.5, the Weighted Averaging method does slightly improve the performance for Next Neighbor Clustering close to the center of the ring. The next dimension to this problem is exploring the effects of varying the number of clusters. A ring of thirty-six nodes was generated with varying numbers of clusters from 3 to 12 clusters. The effects on the accuracy are shown in Figure 4.6.

Figure 4.6 reveals critical information for system design. The number of clusters greatly impacts the accuracy performance of Next Neighbor accuracy. As the number of clusters is decreased, the accuracy increases for a radius of greater than four meters. This shows the number of clusters must be considered before implementing the system for different applications

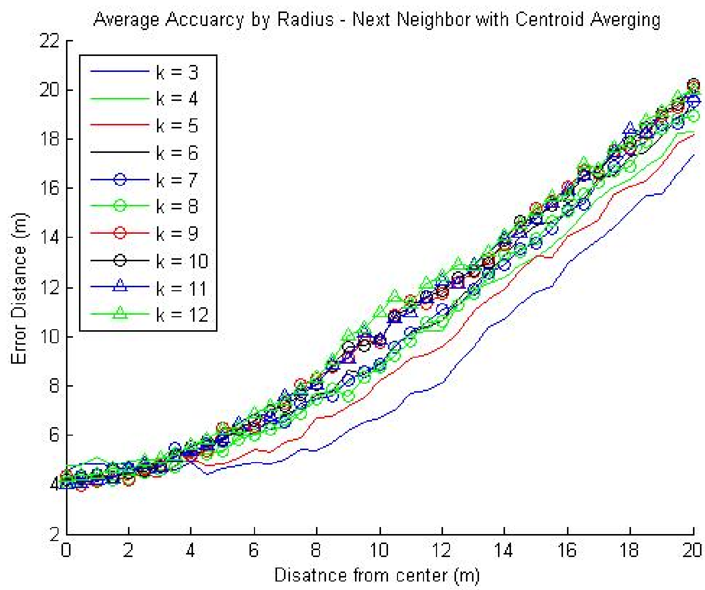

Figure 4.6: **Trials 1 and 2** - Comparison of Different Numbers of Clusters with Next Neighbor Clustering

4.2.3 Node Spacing. For the Node Spacing case, similar experiments were explored as in the Centralized and Next Neighbor case. The results computed are shown in Figure 4.7.

It was found that as the number of sensors in the network increases, the performance of the system improves, shown on the left side of Figure 4.7. This is most noticeable in the region where the radius is equal to the sensor radius. It is interesting that the best performance is the used for base defense, where accuracy close to the sensor network edge is critical. Figure 4.7 on the right, shows that using the Weighted Averaging method reduces performance, and should be further explored and potentially re-evaluated. To explore the Node Spacing results and the effect of varying the number of clusters, a ring with 36 nodes was arranged with varying numbers of clusters. The final comparison of these results is shown in Figure 4.8.

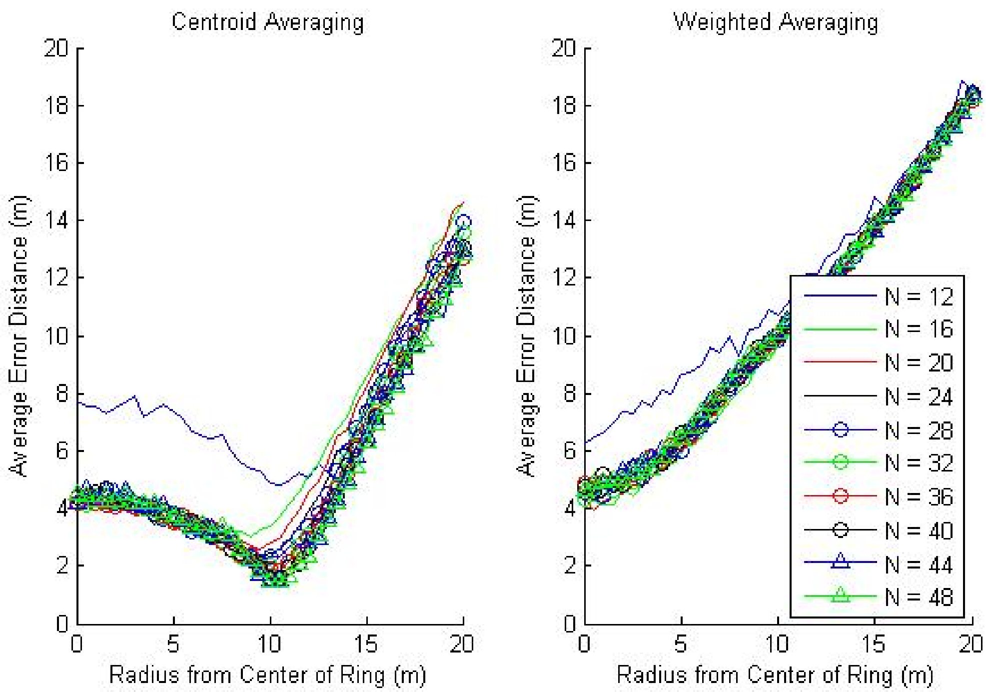

Figure 4.7: **Trial 7** - Comparison of Centroid and Weighted Averaging with 4 Clusters

In Figure 4.8, a result similar to Figure 4.6 is discovered about the effects of changing the numbers of clusters in a localization system. Smaller numbers of clusters lead to improved accuracy for a larger radius, in locations outside of a seven meter radius. The performance is better for larger numbers of clusters inside of the seven meter radius, but becomes worse for a radius greater than seven meters. This is likely related to the fact that fewer large clusters behave like centralized clustering, while more smaller clusters behave less like the centralized case. The exception to this rule is when 12 clusters exist in the network. In this case, the cluster size is three, and additional error is incurred because little data is available.

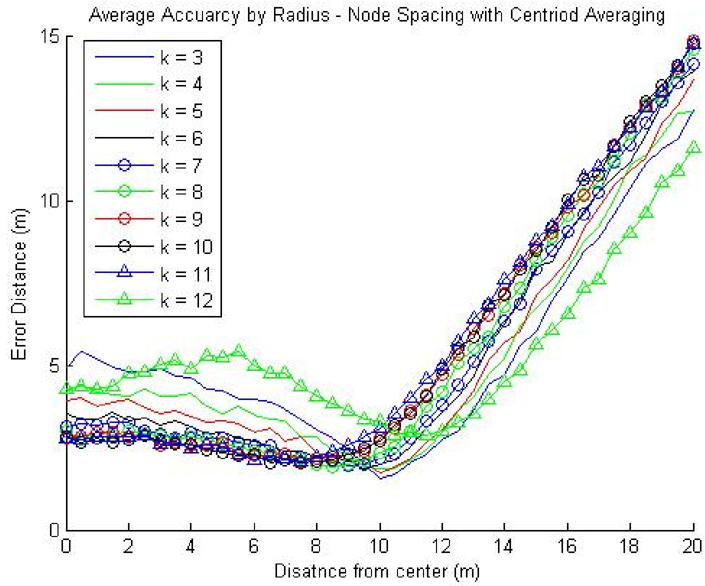

Figure 4.8: **Trial 6** - Node Spacing Accuracy Results for Varying Cluster Size

4.2.4 Analysis. The case of 24 nodes was taken from each of the above simulation cases, and they were replotted here for further analysis.

It can be seen in Figure 4.9 that the Node Spacing Method performs significantly better than the other methods. The next best performer is the Next Neighbor with weighted averaging, followed by Next Neighbor with Centroid fusion, and finally the centralized method. This trend is very important for determining the optimal system for base defense. Notice that the largest gap between the performance of all the systems is at the sensor radius. Here, the Node Spacing method performs much better than the others. For base security, and especially fence line monitoring, the Node Spacing method is the most useful and applicable. This means that in terms of the computational QoS metric, centralized methods are the most effective. This also validates the WDC ideals presented by Datla et al [11]. In WDC, it was shown that

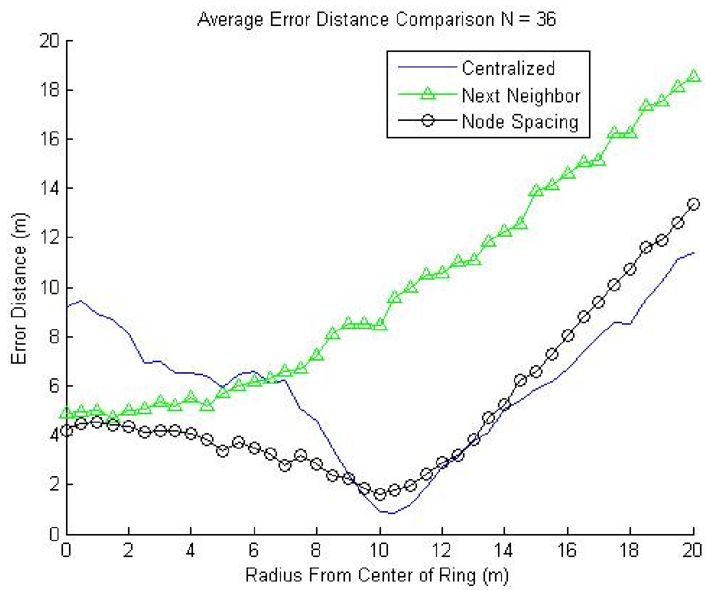

Figure 4.9: Comparison of Methods in terms of Accuracy

introducing distribution can reduce the computational accuracy of the system, and a similar trend is seen here.

To further compare the performance, the average error of each system is computed using Equation (3.10). Then, by comparing these values, Figure 4.10 was generated to show the average improvement created by each system.

In Figure 4.10, it can be seen that for four initiators, the Node Spacing method always performs better. Likewise, the Next Neighbor method is always less accurate than the Centralized method. The average error was then computed by varying the number of initiators on a single sized sensor network. These results are shown in Figure 4.11.

Figure 4.11 shows how changing the number of clusters impacts the performance of the distributed method relative to the centralized method. When there are 11

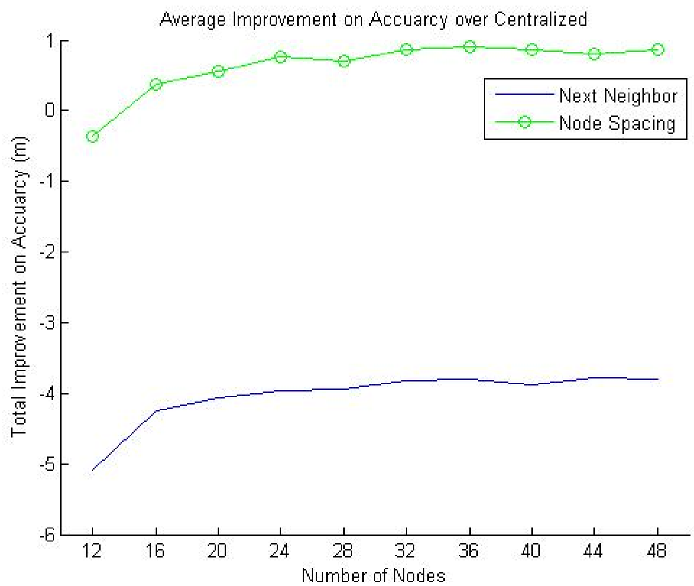

Figure 4.10: Average Improvement over the Centralized Method using Distributed Methods

clusters in the system, the Node Spacing method actually preforms worse on average than the centralized method. The number of clusters in the system provide a potential stumbling point for a system designer. The properties of each cluster size must be taken into account, because increasing the number of clusters negatively impacts the performance of both distributed clustering methods.

Additional tests were performed to examine the impact of adding more nodes to the system while holding the transmitter at a fixed radius. The sensor was held at radii of one and 10 meters. Figures 4.12 and 4.13 show the results from these experiments.

In Figure 4.12, the emitter location is set to be one meter from the center of the ring. This is to simulate asset tracking of a device operating near the center of the base. At the one meter point, it can be seen that the Next Neighbor is the worst

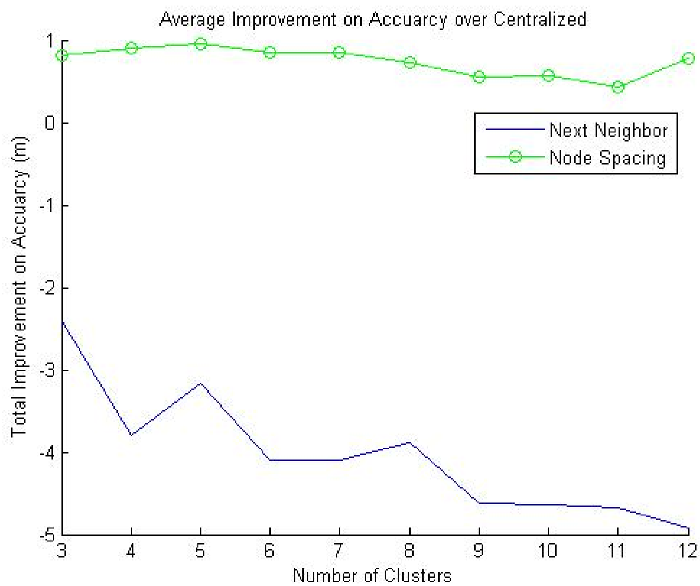

Figure 4.11: Average Improvement over the Centralized Method using Distributed Methods and Varying Numbers of Clusters

preforming method. This may be because of the similarity in sensor location. Another interesting trend is that the Centralized and Node Spacing accuracies improve as the number of sensors N increase.

From Figure 4.13, it can be seen that when the emitter is placed at the same distance from the center of the map as the sensor network, the Centralized method performs the best. Also, the Centralized and Next Neighbor methods appear to remain relatively linear for any number of sensors. There is also a significant amount of fluctuation for different numbers of sensors, which may be caused by the randomness of the system. Another interesting trend is that the Node Spacing and Centralized techniques continues to improve as the number of nodes increases, and the accuracy approaches a minimum error of approximately one meter. This proves that the result is consistent with theory by showing that accuracy improves with additional sensors.

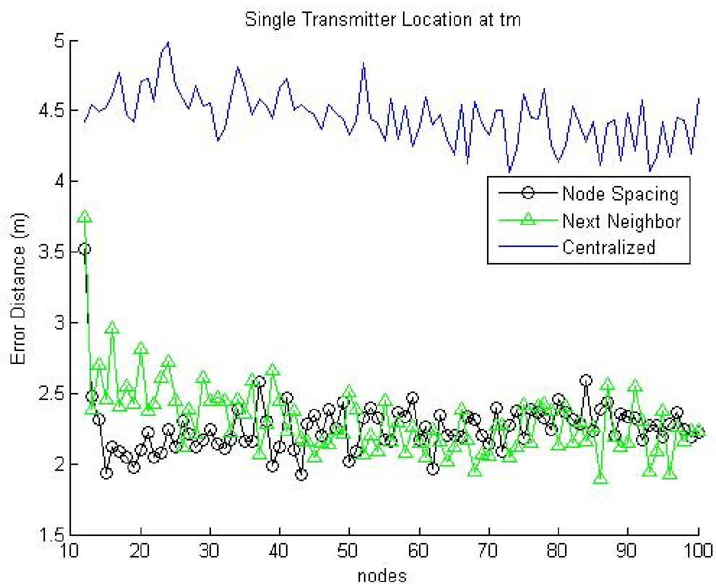

Figure 4.12: **Trials 9, 11 and 13** - Accuracy for Emitter Located one Meter from the Center

In conclusion, the accuracy QoS requirement is shown to remain high for the distributed methods, specifically Node Spacing clustering. The distributed techniques actually perform better than the centralized techniques. There is a small drop-off for WDC techniques in the region outside of the sensor network; however, the RSS localization method used in this research is largely focused on locating sources within a group of sensors. With one of the primary concerns being computational accuracy for WDC, it has been shown here that WDC is effective in localization processes and computational error is mitigated by averaging several measurements.

4.3 Power Results

Power consumption occurs for two primary reasons in a wireless sensor network. The first is the use of power for communication between the nodes. This consumption

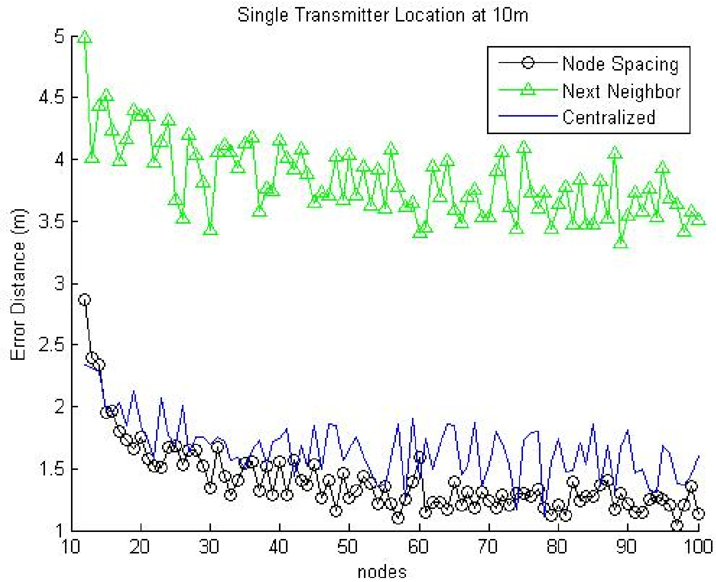

Figure 4.13: **Trails 8, 10 and 12** - Accuracy for Emitter Located 10 Meters from the Center

is largely based on the required communication range and the noise present in the system. The second is the power used while the node is active, which includes idle time and processing of data. These two components can be monitored during the operation of the localization process to better understand which method of localization performs the best in terms of power efficiency. The results for each method are presented and analyzed in the following section.

4.3.1 Centralized. For the centralized case, a large draw of power is expected on each sensor because of the long return distance to the centralized node. This communication distance is the same for each node, so the maximum and minimum power used by any one node is expected to be very similar. The average energy used for all numbers of nodes is the same. This is shown in Figure 4.14 and is compared with the other methods' energy consumption levels. It can be seen that for

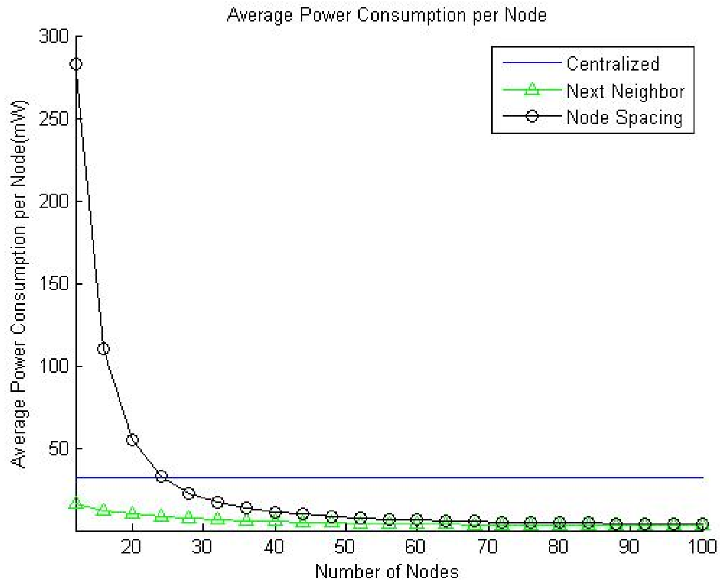

Figure 4.14: **Trials 14, 15, and 16** - Power Consumption for all Methods of Localization

small numbers of nodes, less than 28, the Centralized method performs better than the node spacing method in terms of power consumption. Also, the Next Neighbor method always performs better than the centralized method. This is discussed further in the following sections.

4.3.2 Next Neighbor. The Next Neighbor method is expected to have the best power performance of any of the methods. The results for this experiment are shown in Figure 4.15.

In Figure 4.15, there are several key features to notice. First, the maximum power used by any one node is a constant. The maximum power is always used by the estimating node, which then returns the result of the computation. Second, the minimum and average power consumed both decrease as the number of nodes increase. As the distance between the nodes decreases, the distance required for communica-

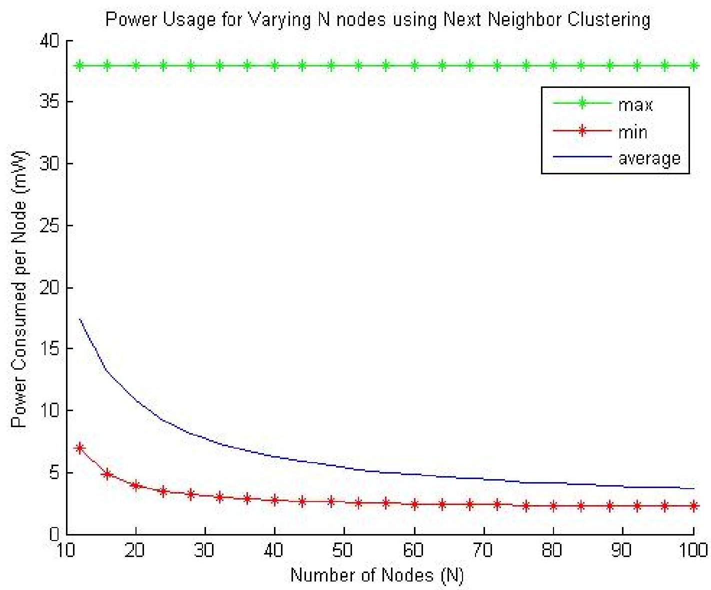

Figure 4.15: **Trial 15** - Power Consumption for Next Neighbor Clustering Method

tion decreases. The reduction in distance lowers the average power expended in the network. Figure 4.14 also shows that this method is the most power efficient method used. In addition to exploring the effects of varying the numbers of nodes, it is also important to see the impact of varying numbers of clusters on the power usage, using an arrangement of 36 nodes, and changing the number of clusters in the network. Figure 4.16 shows the results from this experiment.

In Figure 4.15, as the number of clusters increases, the power required by the system increases. As the number of clusters increase, the size of each cluster becomes smaller. This reduction in cluster size means that more computations occur in the network at the node level, and ultimately this causes the computational requirements to rise.

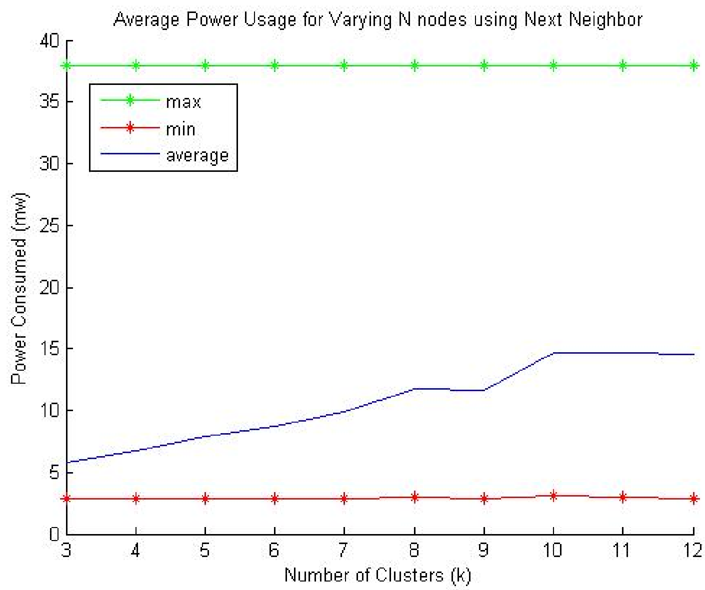

Figure 4.16: **Trial 18** - Power Consumption for Next Neighbor Clustering Method with Varying Numbers of Clusters

4.3.3 Node Spacing. The Node Spacing method is expected to perform better then centralized in most cases; however, performance may be worse in terms of energy consumption for cases with very small numbers of nodes. Large communication distances exist between nodes in a cluster, and this communication distance ultimately creates a larger drain on the available energy of each sensor. There is also a large variance in the maximum and minimum power usage of each node, because of the great variance in distances between each node, and the requirement of the nodes to generate their own estimates. The power results for this clustering method are shown in Figure 4.17.

In the Node Spacing method, there is a large energy demand for networks with small amounts of nodes less than 28 nodes. The transmit distance to the next node in the cluster is further than the distance to the central node. In these cases, the

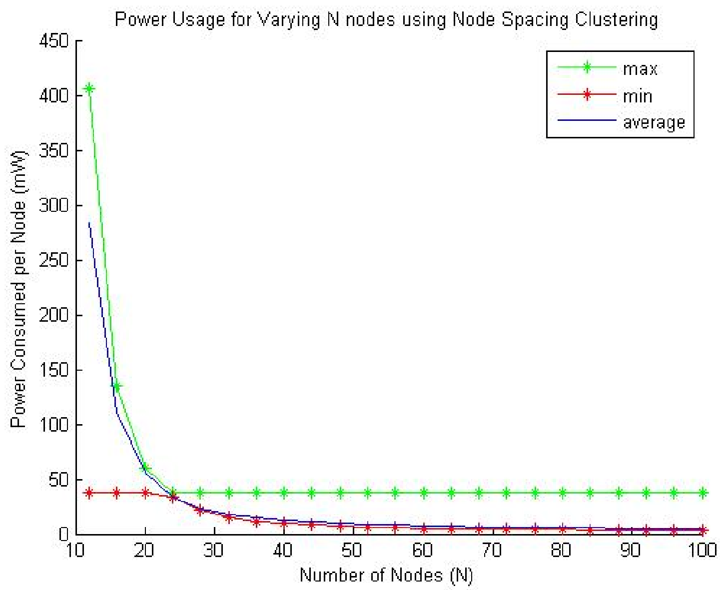

Figure 4.17: **Trial 14** - Power Consumption for Node Spacing Clustering Method

nodes returning the data to the central computer, the least power. Once the distance between nodes in each cluster is less than the distance from the nodes to the centralized node, there is a significant improvement in power efficiency. This transition also causes the maximum power usage to equal the Next Neighbor method. Also, for larger numbers of nodes, the Next Neighbor and Node Spacing methods approach the same value. The distance between nodes becomes negligible as they approach less than a meter in distance.

An additional study was performed on the impact of varying the number of clusters in the network with a ring of size 36 nodes. The results in Figure 4.18 show a trend in Next Neighbor clustering, which is similar to Node Spacing. As the number of clusters increases, the power requirement also increases. This increase is significantly larger than the Next Neighbor method, because as the number of clusters is increased,

both the communication distance and number of estimators increases. This drain is quite high and greatly increases the power consumption of the network.

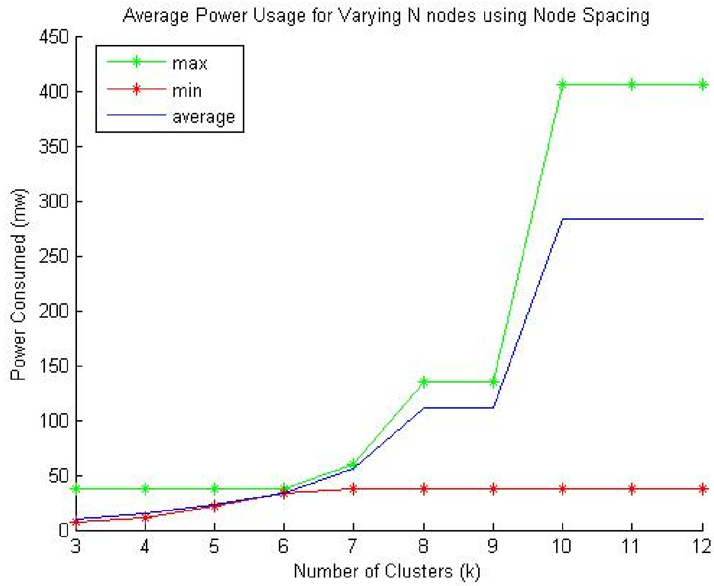

Figure 4.18: **Trial 17** - Power Consumption for Node Spacing Clustering Method with Varying Numbers of Clusters

4.3.4 Analysis. This power analysis is comparable to work performed by Datla in his WDC research. The most important relation is his comparison of network lifetime improvement to the number of nodes. This plot showed around a 94% improved network lifetime [10]. This figure was also reproduced in Chapter II in Figure 2.1. A similar computation was performed on the data collected in simulation and the results are shown in Figure 4.19.

The network life improvement shown here has a maximum life extension of approximately 93%, slightly lower than Datla's theoretical maximum. He showed that 94% could be achieved, with as few as two nodes, while the localization problem needs around 100 nodes to achieve similar gains. Datla et al.'s research assumes that

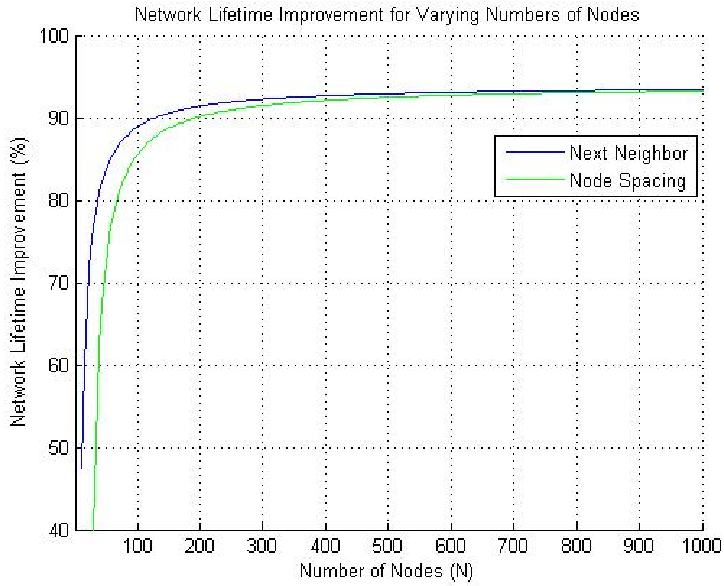

Figure 4.19: Network Life Improvement for Varying Numbers of Nodes

the data collection has already occurred and only requires processing. This thesis work includes the data collection along with the processing, which reduces the efficiency because more tasks are placed on the nodes. The localization data collection can be very power consuming due to the required level of communication needed to share the information to perform the task.

An additional comparison was performed by changing the required computational cost. Different networks and algorithms may have a different ratio between the computational power and the communication power requirements. Three Computational to Communication Ratio's (CCRs) were used in this research for that purpose. A low CCR was used in the previous experiment. Middle and high CCRs are shown in Figure 4.20. The values of the computation cost were changed from 5 mW to 30mW and 50mW, respectively.

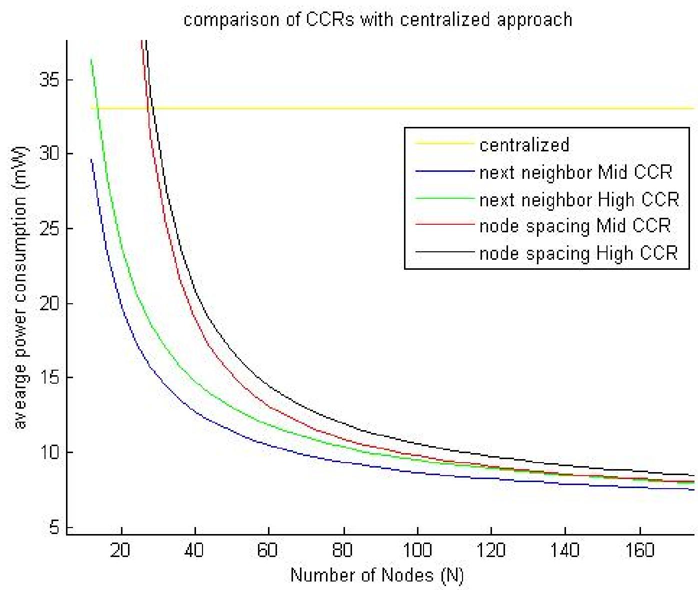

Figure 4.20: Power Consumption Effects using Changes in CCR system Parameters

In Figure 4.20, it is clear that Node Spacing energy consumption is worse than Next Neighbor power consumption. However, the figure does show that if there is a high computation cost, it can be overcome using additional nodes. This indicates that the distributed method can be more energy efficient than the centralized method, as long if there exist enough nodes in the ring network.

In terms of WDC, the expectation is that additional power consumption is required to perform a task in a centralized manner. However, there may be some conditions that would benefit from a centralized method in terms of power. Using a WDC cognitive engine would allow for the network to detect these conditions and determine which method is the most practical. In the results shown, it was found that as the average number of nodes available increased, the average power used in the distributed networks decreased. Also, in the purely WDC method of localiza-

tion, namely Next Neighbor Clustering, the power efficiency was always lower than the centralized technique, due to the optimization of the task graph for power consumption. If the power consumption is not the most critical parameter, and accuracy becomes more important, the cognitive engine may select the Node Spacing method over the Next Neighbor method. Interestingly, as for a large number of nodes, there is little difference in power usage per node for either distributed method. Finally, the network lifetime improvement results are comparable to results computed by Datla et al. [10]. The difference in results is likely caused by the increased computational intensity of both collecting and processing data, rather than assuming it has already been collected. Additionally, the back-haul distance is considerably shorter for this experiment, than in Datla et al.'s, and this shorter distance impacts the overall power performance of the network. In summary, the power consumption QoS metric has been shown to achieve energy improvement over previously explored centralized techniques.

4.4 Time and Bandwidth Results

Time and Bandwidth are two highly related values in this research. As bandwidth increases, the time required to communicate decreases, due to the additional data transfers that can occur simultaneously. This improvement, however, is limited by the number of communications that may occur simultaneously. The lower limit impacts the performance in high bandwidth environments. The following section shows

the impact of bandwidth availability on latency in the network. All three methods are explored and compared.

4.4.1 Channel Definition. A channel or link is a dedicated frequency domain slot of bandwidth. The center frequency this slot may change over time, depending on the number of primary users in the area, and how often the channel is accessed. The network is established with an expected number of available channels at any given time. Channels are made up of a 500 Hz sections of bandwidth, and are used for communications between the different nodes in the network. Each of these channels may also be used to report a final estimate to the central node. It is assumed initially that the central node has a sophisticated receiver which can receive multiple packets over multiple channels simultaneously. The 500 Hz bandwidth requirement was determined according to Figure 4.21.

It can be seen in Figure 4.21 that a delay of 0.284 seconds is incurred at the highest latency, for a packet containing 20 measurements with a 1/4 code rate. This will likely by more than enough coding to prevent correct bit errors, while also supporting and above average cluster size. This delay is not significant enough to be impractical when compared to the time required for the localization estimate. Here, a code rate of 1/2 would be more than satisfactory for accurate communication, and to prevent any significant bit errors.

4.4.2 Centralized. For Centralized processing of the collected RSS data, it is expected that a large amount of bandwidth will be required to prevent any

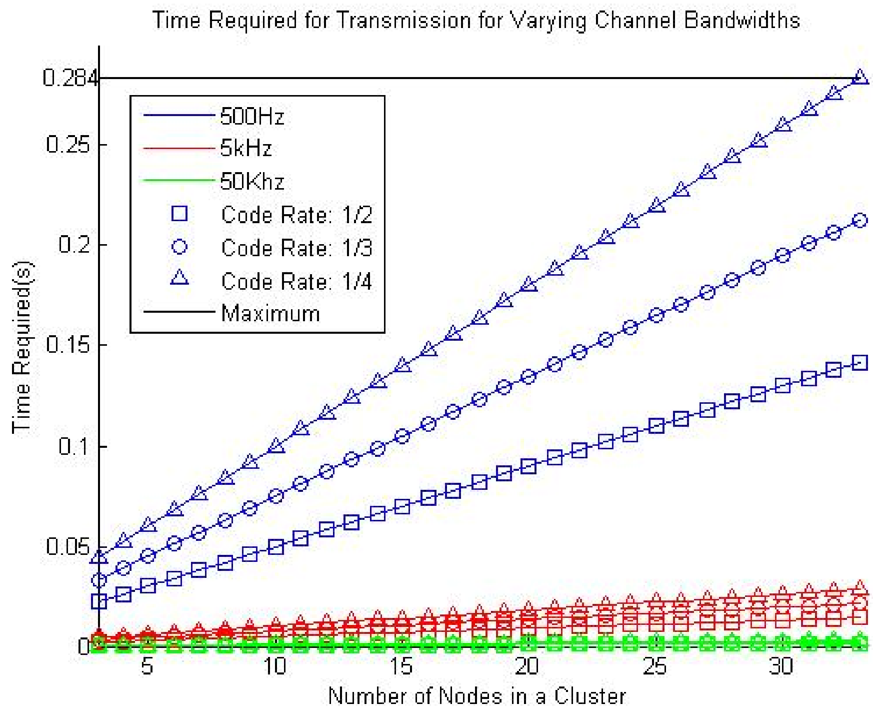

Figure 4.21: Predicted Communication Latency for Varying Code Rates and Bandwidth Usages

interference from occurring. Otherwise, the nodes will have to wait until the channel is unoccupied before returning their RSS measurements. The results for the Centralized bandwidth availability and time constraints are shown in Figure 4.22. The derivation of the equation used for Figure 4.22 is found in Appendix III, the final result is reproduced in Equation 4.1. The Centralized case has a very high time delay when the radio environment is highly utilized; however, there is rapid time improvement with additional bandwidth channels for communication. The system peaks in terms of time performance, when the number of channels equals the number of nodes in the system. Here, one time step is needed to return all of the collected data, and one time step is needed to process the data.

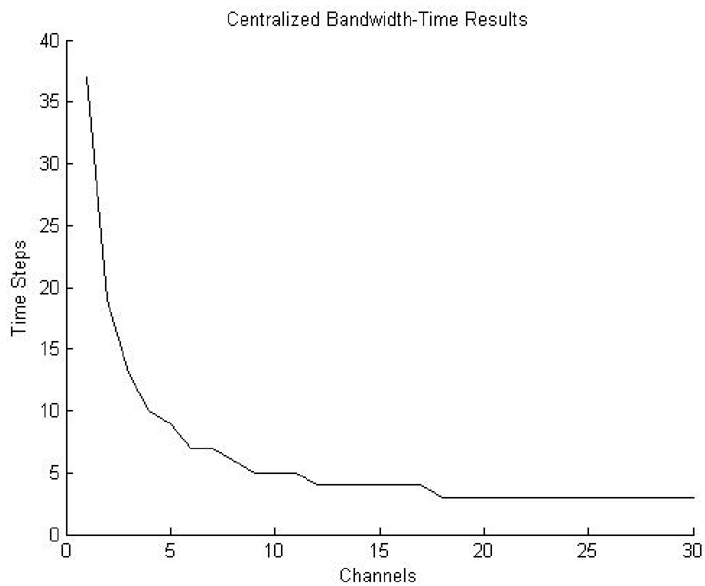

Figure 4.22: **Trial 20** - Centralized Bandwidth-Time Results for 36 Nodes

The derivation of the equation used for Figure 4.22 is found in Appendix II, the final result is reproduced in Equation (4.1), where t is the time steps required, N is the number of nodes in the network, and c is the number of channels available.

$$t = \left\lceil \frac{N}{c} \right\rceil + 1 \tag{4.1}$$

4.4.3 Distributed Methods. For the Distributed methods, a derivation for the number of time steps required is given in Appendix II. The final result for the Next Neighbor case is given in Equation (4.2), where n is the number of nodes in each cluster, k is the number of clusters. Channel reuse is possible, because each cluster can communicate internally without interfering with other clusters.

$$t = n + \left\lceil \frac{k}{c} \right\rceil \tag{4.2}$$

The Node Spacing case has a more complicated results, based on the number of clusters. If sufficient clusters are used, then channel reuse is possible. If not, then only some of the clusters may communicate simultaneously. If $N \geq (2k+1)k$ then Equation (4.2) can also be used. If $N \leq (2k+1)k$, then Equation (4.3) must be used, where $k_m ax$ is given by Equation 4.4 and represents the maximum number of clusters that may communicate on the same channel. Additional details can be found in Appendix II.

$$t = \left\lceil \frac{k}{k_{max}c}(n-1) \right\rceil + 1 + \left\lceil \frac{k}{c} \right\rceil \tag{4.3}$$

$$k_{max} = \lfloor n/2 \rfloor + 1 : N < 2k^2 + k \tag{4.4}$$

In the Distributed cases, the limiting factors are the number of clusters and cluster size. As the number of clusters increases, more bandwidth is needed to return the estimates to the centralized node. However, if the number of clusters is reduced then the cluster size must increase. As the cluster size grows, additional hops are required, and there is additional latency incurred, which cannot be alleviated with additional bandwidth. Figure 4.23 shows the results from the Distributed methods when compared to the Centralized method.

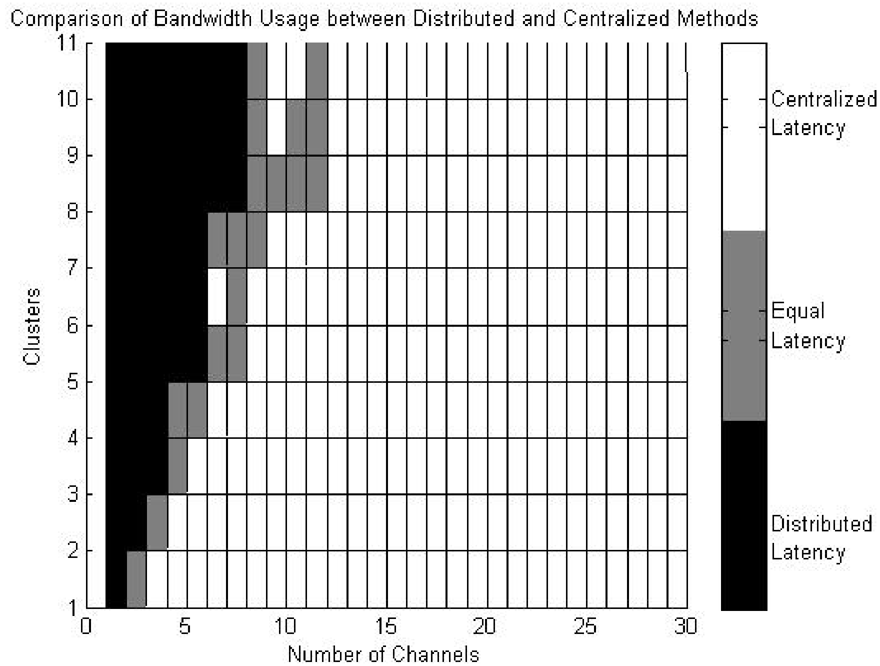

Figure 4.23: **Trial 21 and 22** - Bandwidth-Time Plot Comparing Distributed and Centralized Methods

In Figure, 4.23 the black regions in the upper left corner indicates the regions where Distributed methods require less time to perform the localization with a fixed number of channels available. The gray regions indicates where both Distributed and Centralized methods perform equally as well. Finally, the white regions show the areas where the Centralized methods performs the best under the given bandwidth conditions. Low bandwidth availability and large numbers of clusters benefit the Distributed methods. There are also an few interesting points in the equal latency regions, where the centralized method preforms better. This is caused by cluster interference when the nodes are reporting estimates back to the central node. Additional bandwidth improves this performance for the decentralized cases.

4.4.4 Analysis. Bandwidth and latency of a localization algorithm are critical to understanding the value of the system. Figure 4.24 shows a comparison of Next Neighbor and Node Spacing clustering.

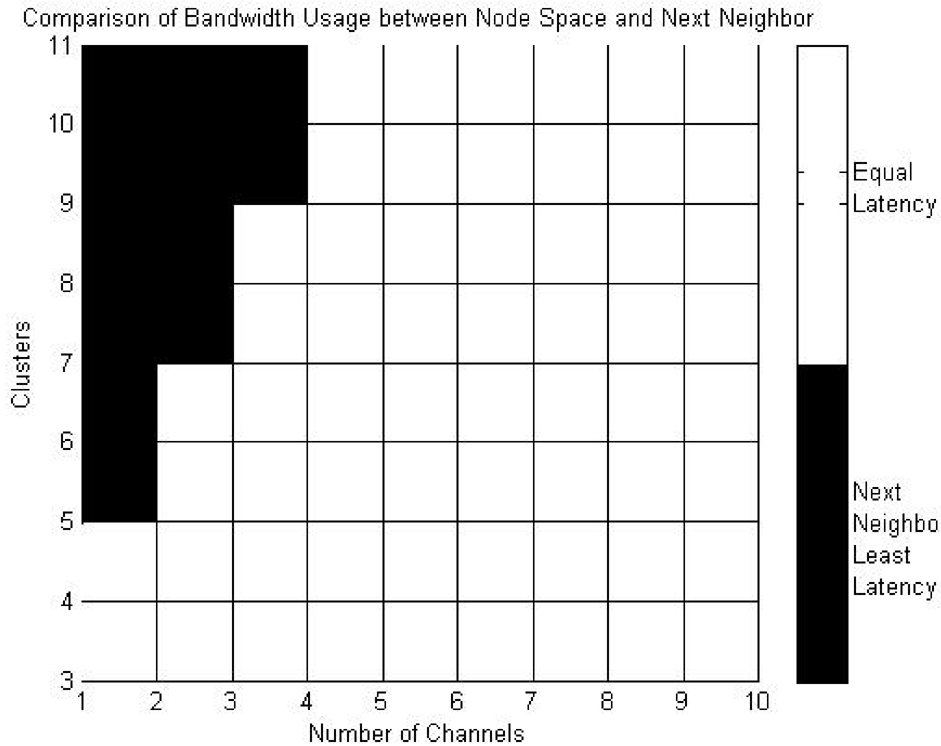

Figure 4.24: Comparison of Systems with 36 Nodes

Figure 4.24 shows that Next Neighbor always preforms equal to or better than Node Spacing. The black regions represent where Next Neighbor requires less time to complete the localization with the available bandwidth, and the white regions represent where they require and equal amount of time. For an overwhelming majority of the cases, both methods perform equally, and the bandwidth metric can be ignored when selecting a localization method.

The overall conclusion from this set of simulations is that if there is low bandwidth, both of the distributed methods preform well. Next Neighbor guarantees the optimum performance in those low bandwidth environments, while Node Spacing may be the same or worse than the Next Neighbor method. When large amounts of bandwidth are available, the Centralized method performs better in terms of time efficiency.

4.5 Key Findings

The three QoS metrics explored in this research are the Computational Accuracy, Power, and Latency. These metrics were explored as a tradespace, and sacrifices must be made to select the best solution for any given situation. The findings show that for the ring topology, representative of a series of sensors along a fence line, WDC could be used in a meaningful and advantageous system. The overall results, shown in Table 4.1, demonstrate the tradespace for each requirement.

Table 4.1 gives a general synopsis of the final results from each of the QoS metrics explored. Using this table, it is possible to identify which method is the most effective for a given situation. Because of the accuracy advantages of Node Spacing, it would be most useful for asset tracking of objects on a base. Next Neighbor provides high energy conservation, but its weakness in localization makes it somewhat impractical for RSS estimation. It could be adapted for cooperative spectrum scanning and spectral mapping of the radio environment in near real time. Centralized has the greatest advantage for its accuracy outside of the ring; however, the accuracy is still

relatively poor. It could be used in cases where large amounts of available bandwidth exist.

4.6 Conclusion

Wireless Distributed Computing can be applied to many different tasks involving cognitive radios. Its application in localization has been shown to be a practical approach, which can improve computational accuracy in certain situations. It significantly improves power consumption and can extend the battery life of a sensor network by nearly 100%. Finally, there are bandwidth improvements that can be obtained from applying WDC in an intelligent manner to avoid interference with other radios. There may be initial setup needed for the radios to establish communication links, and discover the optimal routes for communication. Once this is established, there are significant energy and bandwidth gains to be made, which can overcome the initial configureation overhead.

Table 4.1: Trade Space Results

Requirement	**Control**	**Experimental**	
	Centralized Mean	**Next Neighbor** Mean % Improvement	**Node Spacing** Mean % Improvement
Computational - Accuracy 0-4 meters	Worst 8.05m	Average 5.45m 32.4%	Best 4.54m 43.6%
Computational - Accuracy 4-9 meters	Average 5.48m	Worst 6.98m -26.16%	Best 3.58m 34.68%
Computational - Accuracy 9-20 meters	Best 5.85m	Worst 13.45m -129.94%	Average 6.94m -18.80%
Power Energy Consumption - 12-20 nodes	Average 33.0mW	Best 13.75mW 58.33%	Worst 149.86mW -354.13%
Power Energy Consumption - 20-60 nodes	Worst 33.0mW	Best 6.41mW 80.56%	Average 14.69mW 55.47%
Power Energy Consumption - $\geq$ 60 nodes	Worst 33.0mW	Best 2.49mW 92.44%	Best 2.75mW 91.65%
Latency - 1-5 Channels	Worst 17.6 Time Steps	Best 9.75 Time Steps 44.60%	Average 11.1 Time Steps 36.93%
Latency - 5-10 Channels	Best 6.5 Time Steps	Worst 7.90 Time Steps -21.58%	Worst 7.93 Time Steps -22.01%
Latency - 10-30 Channels	Best 3.47 Time Steps	Worst 7.51 Time Steps -116.10%	Worst 7.51 Time Steps -116.10%

V. Conclusions

This section provides a summary analysis of the data presented in Chapter IV. The results are reviewed to make recommendations on how the system may be employed, and for what purposes they may be optimal. Finally, suggestions for future work will be presented that could continue this area of research.

5.1 Trade Space Analysis

This section provides an analysis of the trade-offs associated with selecting each method of clustering.

5.1.1 Centralized. The following Pros and Cons lists represent the compiled results for Centralized Localization.

Pros:

- Most Accurate Method Outside of the Network Area – When the transmitter is further than nine meters from the center of the network, it was found that the centralized method was found to be the most effective.
- Low Variability of Node Energy Usage – All of the nodes in the centralized case have the same responsibilities and use the same amount of power.
- Minimal Latency with High Bandwidth Availability – If there is a large amount of bandwidth available for secondary users, the Centralized method has the least latency, since only one hop is required to generate an estimate.

Cons:

- Poor Accuracy near the Center of the Network — Both distributed methods of localization perform better when the transmitter is within four meters of the network center.
- High Average Power Usage per Node — The average power usage for the centralized case was found to be around 35 mW, which is higher than the average energy usage for most of the distributed configurations.
- High Bandwidth Requirements — If there is a small amount of bandwidth available, there is high latency with the Centralized case.

The Centralized method of localization was used as a baseline and describes the current method of collecting data for localization processing. Centralized localization has both several advantages as well as drawbacks. The Centralized method was the most accurate method outside of a nine meter radius. This comes at a cost of performing worse inside the nine meter area, than both of the distributed methods. The Centralized method maintains a constant power requirement, related to the power required to return a RSS value to the central node. This means that power usage per node is the same regardless of the number of sensors in the network. In the case of a small number of sensors, Node Spacing clustering loses power efficiency, because of the increase in required transmission distance. If there are large numbers of nodes available, typically more than 24 nodes, distributed methods are more power efficient. Finally, Bandwidth and Time are directly related: if more bandwidth is available, less time is required. The distributed methods use bandwidth more efficiently and require less time if fewer channels are available. If there are large numbers of links

or channels, then the Centralized methods out performs the distributed methods in terms of bandwidth.

5.1.2 Next Neighbor. Based on the analysis from previous chapters, the following pros and cons were found for the Next Neighbor Method of Localization.

Pros:

- Good Accuracy at Network Center — The accuracy at the center of the sensor network is the same as Node Spacing and better than Centralized.
- Maximizes Energy Usage per Node — This method maximizes the average energy usage for the network.
- Excellent Bandwidth Efficiency — This clustering setup enables multiple clusters to communicate simultaneously using links with the same bandwidth.

Cons:

- Poor Overall Accuracy — The accuracy steadily deteriorates as the transmitter gets further from the center of the network, and this method becomes the worst in terms of accuracy.
- High Maximum Node Energy Usage — Because one node per cluster is responsible for both the estimate and communicating to the user, there is a higher power draw on this node.

- High Latency Requirements — Because this clustering method requires a multiple hops before an estimate can be made, there is an increased minimum latency for the network.

The Next Neighbor method of clustering was designed to reduce the power footprint of the localization process. The clustering method does this by wisely selecting clusters to limit the communication range between nodes. The results from Chapter IV showed that a large amount of energy savings can be obtained by using Next Neighbor method of localization. Specifically, gains of over 60 percent for small numbers of nodes can be gained over the Centralized method. However, this did come at a cost—there is a significant loss of accuracy outside of five meters compared to the Centralized method. This loss in accuracy would make it difficult to use this as a practical method of localization. One advantage is that the accuracy performance is not affected by the number of nodes in the network. In terms of bandwidth usage, when there are fewer available channels for communication, there are fewer time steps required to complete the localization process. This is the result of multiple clusters being able to communicate simultaneously. It was found that Next Neighbor performs better than Centralized metrics of power consumption and bandwidth latency metrics, but is weak in the area of accuracy.

5.1.3 Node Spacing. The following are the generalized pros and cons for Node Spacing:

Pros:

- Very Accurate Inside Sensor Network — This method performs very well for a transmitter less than nine meters from the center of the network.
- Energy Efficient for Larger Sensor Networks — As more sensors are added to the network, this method of localization approaches the energy efficiency of Next Neighbor Localization.
- Excellent Bandwidth Efficiency — For low bandwidth availabilities, it can still maintain a relatively low latency when compared to the Centralized method.

Cons:

- Accuracy Becomes Poor Outside Sensor Network — The performance of the method is significantly reduced outside the sensor network.
- High Energy Requirements for Small Sensor Networks — The power used with a smaller number of sensors is larger than the Centralized method's power usage.
- High Latency Requirements — Similar to the Next Neighbor method, the latency is dependent on the number of hops required for the clusters to communicate.

Node Spacing localization was designed to optimize data collection by spreading out sensors within the same cluster. The spacing between nodes allows the data collected to be less correlated and the final estimates to be more accurate. It was found that, within a radius of nine meters, Node Spacing clustering was more effective in terms of accuracy, and the results between Centralized and Node Spacing are similar beyond the 12 meter point. Node Spacing provides the most accurate method of

localization. The accuracy of the Node Spacing is highly dependent on the number of clusters and cluster size in the network. Changing the clustering impacts the shape of the accuracy curve, and affects how accurate the system is at two points: at the center of the network, and at the sensor radius. Additionally, in terms of power usage, Node Spacing performs better than centralized when there are more than 24 nodes in the system, and approaches the energy usage of Next Neighbor clustering as the number of nodes increases. This shows that Node Spacing is the average performer of the three methods in terms of power consumption. Finally, in regions with low bandwidth, the Node Spacing method performs better in terms of latency than the Centralized method. However, it may not perform as well as the Next Neighbor method when cluster sizes are small.

5.1.4 Conclusions. Each method has its merits as a useful technique for localization. Each has areas where it performs best, and it is up to a system designer to choose which method to apply. One key trend could help a system designer with this task, that is, how cluster size and number of clusters affect all of the metrics. This research sought to answer several research questions, one of which being: can WDC be paired with RSS to successfully localize an active transmitter? This research shows, definitively, yes there are ways to use WDC to maintain the accuracy of localization, while still having gains in terms of power and bandwidth. Additionally, a second research question was to provide a trade space analysis for the various methods explored. This was also performed and presented previously. The final research ques-

tion was: which situations would each method be best applied to? This is explored in Section 5.3.

5.1.4.1 The Effect of Cluster Size. Each metric was greatly affected by the cluster size, which was inversely related to the number of clusters in a network. In terms of accuracy, smaller cluster sizes improved accuracy near the center of the search grid, and reduced the accuracy further from the center. Likewise, larger cluster sizes improved the accuracy for larger radii and reduced the accuracy at the center of the network. Additionally, increasing cluster size reduced the average power consumption of the system, while decreasing cluster size reduced the power consumption. The final metrics of Bandwidth and Latency also showed that reducing cluster size reduced the bandwidth needed to maintain latency. In summary, cluster size is a key concern for a system designer and must be considered when selecting a distributed clustering method.

5.2 Contributions

The research area explored in this thesis sought to explore and further the basic knowledge of Wireless Distributed Computing by applying it to localization, specifically using RSS and the ML algorithm. WDC previously had only been applied theoretically to processing already collected data. The localization problem focuses on both processing and collecting data in real time. The application of WDC was shown here to improve power and bandwidth performance of the system network, and even improve the computational accuracy of the localization process. This research

validates WDC's effectiveness when applied to localization. The following contributions represent the success of this research project.

- Two methods of applying WDC to localization were determined.
- Accuracy, Power Usage, and Bandwidth and Latency metrics where all measured.
- A simulation environment was developed for testing different communication and task graphs on the maximum likelihood estimation algorithm with WDC applied.
- A trade space analysis exploring the impact of WDC on RSS localization was generated.
- Potential methods of system deployment were discussed.
- The initial localization results on CORNET test bed were inconclusive due to the complexity of the propagation model in the building (see Appendix D).

5.3 Deployment Recommendations

Each method of clustering has its own advantages and disadvantages. The Centralized method has the greatest advantage of accuracy near the edge of the sensor network, since it operates quickly with high a bandwidth availability. The Centralized method could be useful for a base security system and the monitoring of active transmitters moving near the perimeter of a base. This method is very accurate near the edge of the sensor network, and ideal for a fence line. The method could be used

to tell if there is someone crossing the base perimeter. Figure 5.1 shows an example of how Centralized Localization can be applied to a real world situation.

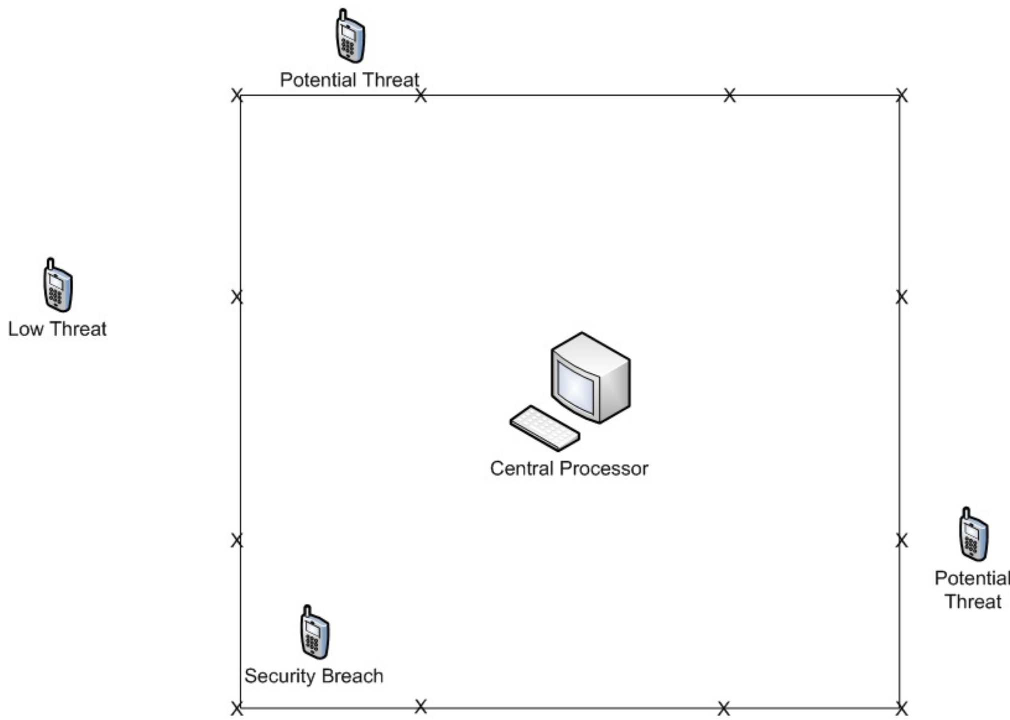

Figure 5.1: Potential Centralized Use

In Figure 5.1, the cell phone icon indicates an active transmitter located in the area. Each transmitter is labeled with a threat level indicating how likely a threat the device's user may be. The x's along the black line indicate sensors place along a fence line at a FOB. The central processor receives all of the RSS measurements from these sensors and uses the information to determine the devices' locations and the users' threat levels. The Security Breach in the lower left corner can then be reported to security forces so they may respond.

Next Neighbor clustering is better in terms of energy efficiency when compared to the other methods of localization explored. This method has poor accuracy which

makes it difficult to deploy. However, the method may be best employed as a cooperative spectrum sensor to avoid interfering with communication channels, or to identify potential channels for jamming. This could be a method for quickly identifying channels for DSA while minimizing overhead on secondary users. Figure 5.2 demonstrates how Next Neighbor clustering could be applied.

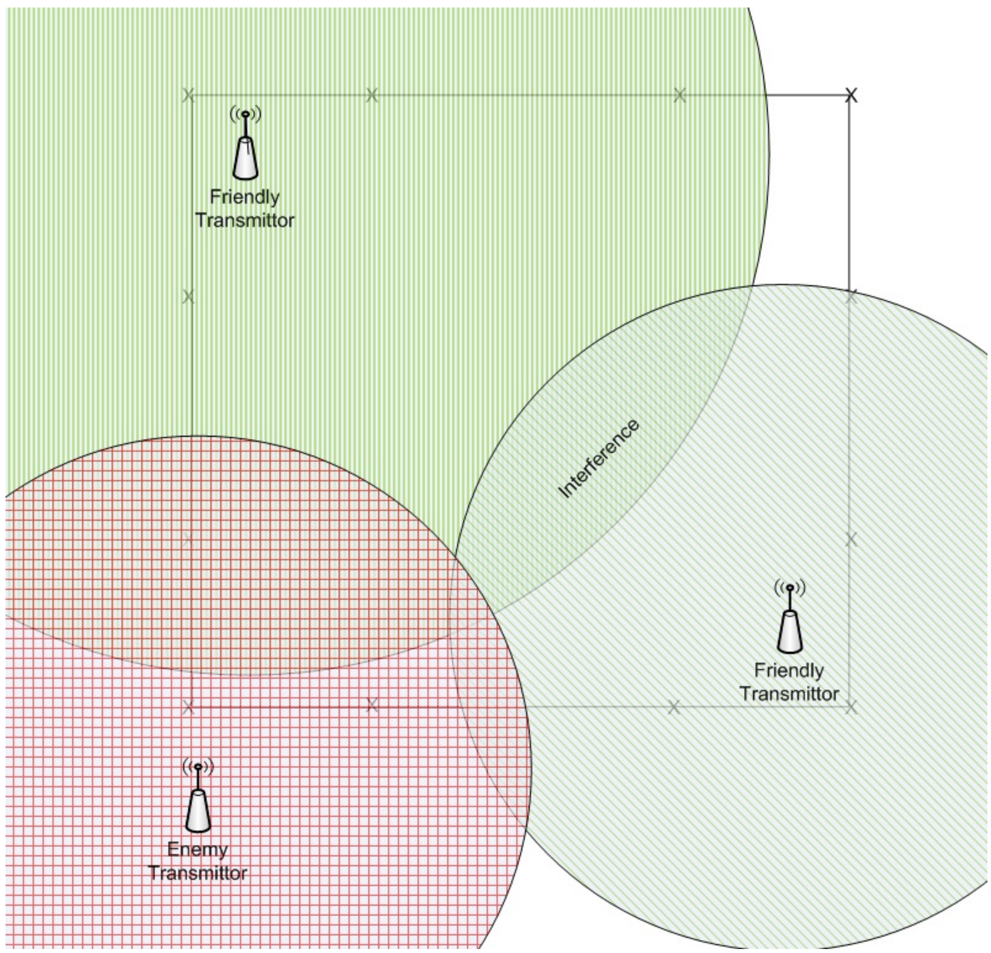

Figure 5.2: Potential Next Neighbor Use

Figure 5.2 shows how the sensor network could be used to detect potential interference as well as enemy transmission frequencies. The sensor network can allow a jammer to cognitively jam the enemy signal without affecting the friendly signal.

Also, it is possible to predict the friendly transmission interference, so the potential problem can be avoided. The map shows where all friendly transmitters are located and their projected transmission patterns. This is often estimated by a communication officer and not measured. The application of WDC localization can allow for more accurate spectrum usage maps to be generated.

The Node Spacing method provides good accuracy within the sensor network, while limiting the energy and bandwidth footprint. This could be directly applied to asset tracking on a base because of its accuracy within the ring. By associating an active transmitter a human using a device like a cell phone or walkie talkie, it would be possible to track them throughout the base. This could be done to monitor them for security as well as safety reasons. Figure 5.3 shows a projected usage for the Node Spacing method of localization.

Using the Node Spacing clustering it is possible to estimate where an active transmitter is located. Figure 5.3 represents three main areas on base. The first area is a safe place for base personnel; the dining hall is represented in green. The yellow area represents a flight line where there is a potential safety concern. There is one individual who has wandered onto the flight line. Next, there is a secure area. This could be a secure information vault, weapons depot, or hanger. When someone enters this area with an unregistered active transmitter, they must be detained by security forces.

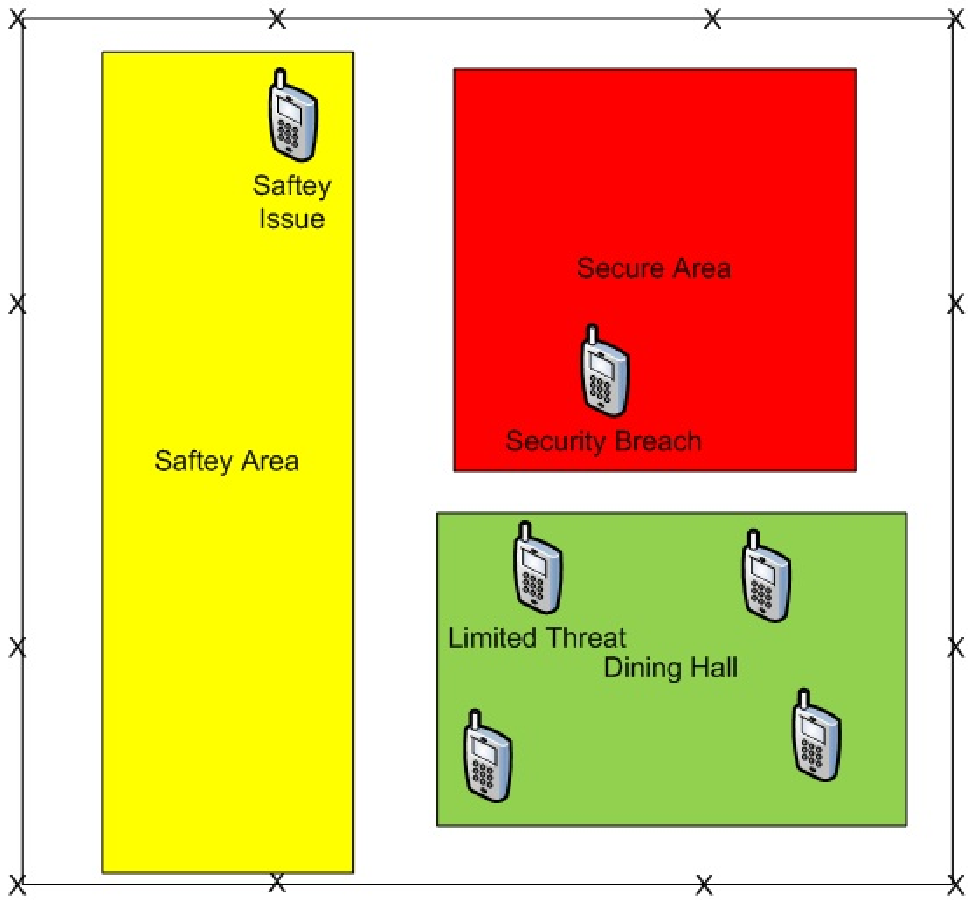

Figure 5.3: Potential Next Neighbor Use

5.4 Future Works

There is still a large amount of work to be done in the area of Wireless Distributed Computing and Localization. First, research must go into improving the realism of the current model. This could be done by first removing the assumption of a fully connected task graph (see Appendix V for initial results). Next, introduction of packet collisions and the impact of dropped packets on accuracy must be factored in (see Appendix V for initial results). Next, different topologies may be explored, one such example is the a grid of sensors (see Appendix V for initial results). Additionally,

the work done here was performed exclusively in simulation and can still be applied to a real world system either using USRPs or another programmable cognitive radio device. By testing some of these principles on a system, the simulation can be validated and improve the understanding of WDC. Another area that can be expanded is incorporating antenna pattern detection from the ML Estimation algorithm. Using the antenna pattern capability would allow for accurate spectrum usage maps to be generated based on the antenna patterns of the transmitters in the search area.

5.5 Concluding Remarks

This thesis presented potential methods of improving centralized localization techniques by applying WDC. Each method was evaluated in terms of three metrics: accuracy, power consumption, and bandwidth and latency. The results from these experiments were recorded and compared for use by a future system designer. It was shown that each of these methods have advantages and disadvantages, and could be applied to a wide verity of scenarios. Additional research must be explored to fully develop a distributed localization system and deploy it into a real world environment.

Appendix A. Verification of Independence on Angular Location in Localization

In order to verify that a polar source movement approach was valid, it was necessary to show that the accuracy was statistically independent from the angle of the source. To prove this, 95% confidence intervals were placed around each of the simulation results for a constant radius, and the angle was varied from 0 to 2π radians steps of $\pi/2$ radians. Since all of the curves fell within their confidence intervals, it can be said that simulation events are statistically similar.

A.1 Centralized

The first simulation case that was checked used the Centralized Method of localization. The plot shown in Figure A.1 gives an example of the collected results for one case containing 48 nodes.

It can be seen in Figure A.1 that the vast majority of points fell within each other's confidence intervals. This shows that the expected results for each θ is statistically independent. There was only one outlier at the radius of 10 meters. Notice that this one point was slightly higher than the other points, and it was not encompassed by the confidence intervals of the other points.

A.2 Distributed - Centroid Method

The process of determining independence with respect to angular placement was also performed for the distributed cases. This was done by varying θ with steps of $\pi/2$

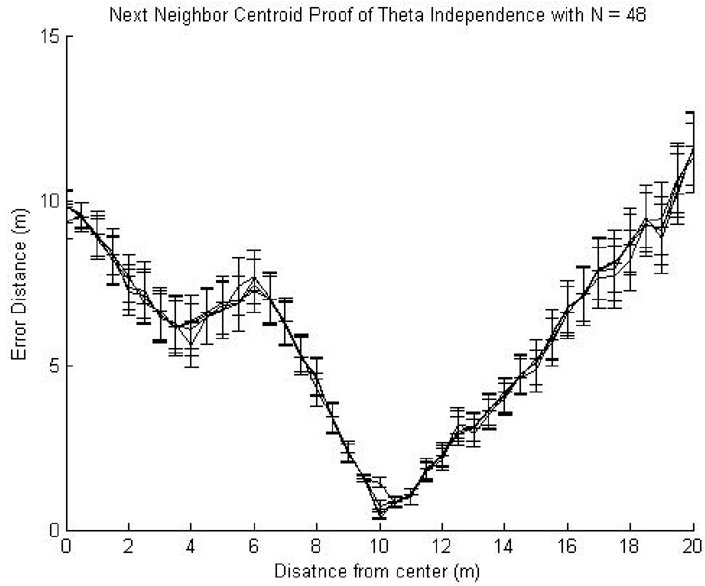

Figure A.1: Average Accuracy vs Radial Distance for Centralized Case

radians, around one full rotation, from 0 to 2π radians. Since all points fell within the 95% confidence intervals of each other, the collected data points were determined to be independent of θ. Therefore, the final results were averaged together for comparison. The results for the Next Neighbor and the Node Spacing methods, using the Centroid method, are shown in Figures A.2 and A.3. Because all of the estimates are in the selected confidence intervals, they are independent of θ.

A.3 Distributed - Weighted Average Method

Findings show that the Weighted Average Method is only independent of angular location, when using the Next Neighbor method. Figures A.4 and A.5 show the test for independence.

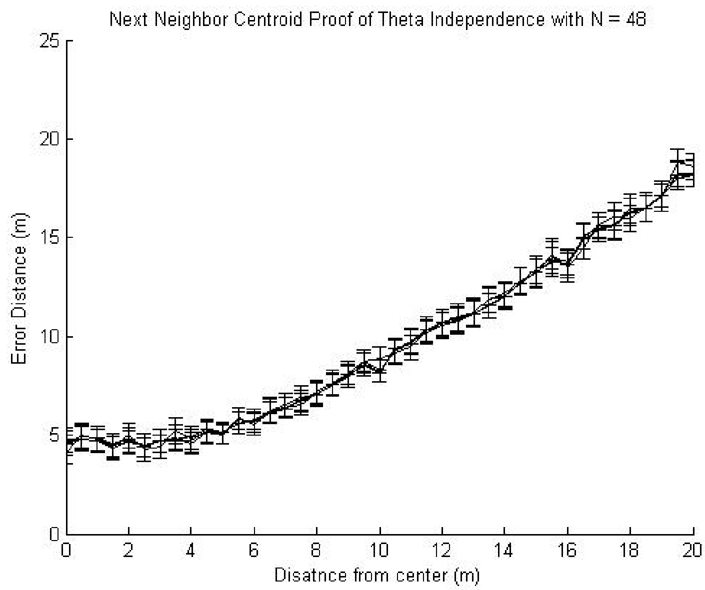

Figure A.2: Results of Localization for Varying θ for Next Neighbor using Centroid Estimate Fusion

Notice that in Figure A.5, the curves are no longer restricted to each other's confidence intervals after a radius of three meters. This means that after this point, the localization becomes a function of the angle.

A.4 Conclusion

It was found that the results are independent of θ for the majority of cases, but not for the case of Weighted Averaging with Node Spacing. The final results are tabulated in the Table A.1

Table A.1: Summary - Statistical Independence of θ

Clustering	**Averaging**	**Independence Region**	**Dependence Region**
Centralized	N/A	0-20 meters	none
Next Neighbor	Centroid	0-20 meters	none
Next Neighbor	Weighted	0-20 meters	none
Node Space	Centroid	0-20 meters	none
Node Space	Weighted	0-1, 9-12 meters	1-9, 12-20 meters

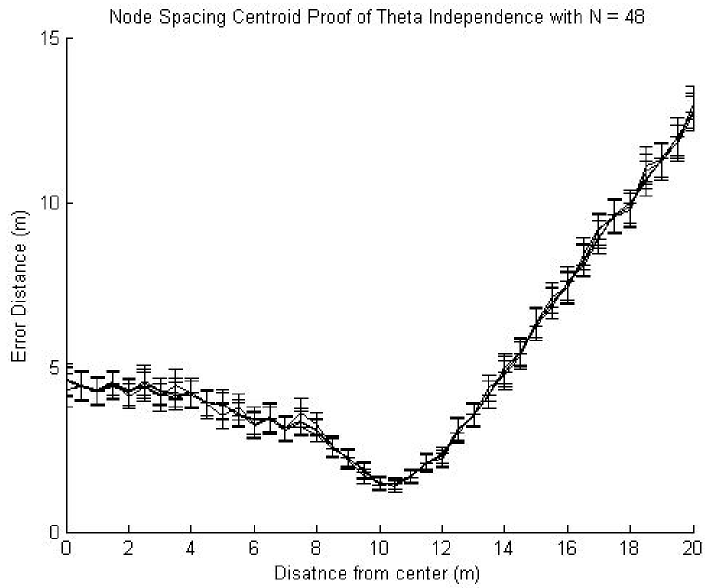

Figure A.3: Results of Localization for Varying θ for Node Spacing using Centroid Estimate Fusion

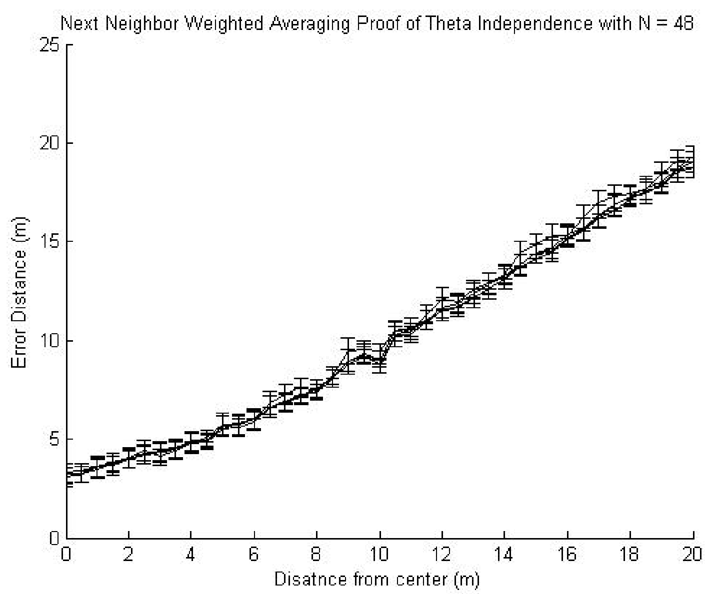

Figure A.4: Results of Localization for Varying θ for Next Neighbor using Weighted Averaging Estimate Fusion

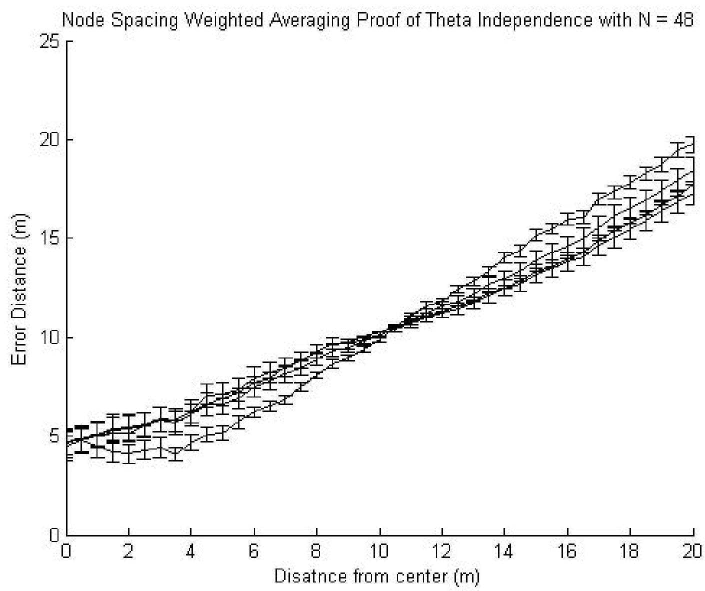

Figure A.5: Results of Localization for Varying θ for Node Spacing using Weighted Averaging Estimate Fusion

This section explains in detail the derivations for how the bandwidth and time metric equations were developed. It also provides simulated background for the results. The variables in Table B.1 are used in the derivations.

Table B.1: Derivation Variables

Variable	**Meaning**
N	Number of Nodes in the Network
n	Number of Nodes in a Cluster
k	Number of Clusters in the Network
t	Number of time steps
c	Number of Bandwidth Channels
r	Radius of the Sensor Ring
d	Required Communication Distance

Three assumptions are made in the derivations. First, each channel provides the required communication bandwidth over a given time interval, equal to the time steps. Second, these channels are adaptively selected to avoid other primary users in the region. Third, there is no need for data retransmissions. Retransmissions are handled using short packets, which are coded with a minimum of a 1/2 code rate code.

B.1 Centralized

The first method explored for bandwidth purposes, is the Centralized Method. The Centralized Method requires that all sensor nodes send a single packet to the user, and the estimation be performed locally on the user's machine. The Centralized Method, therefore, has a high bandwidth requirement in order for in the system to operate in synchronous fashion. A graphical representation of the system interfering

in communication is shown in Figure B.1. The multicolored rings represent the transmitted signals from a selection of nodes. The nodes' signals overlap, signifying that there will be interference, if the nodes transmit over the same channel.

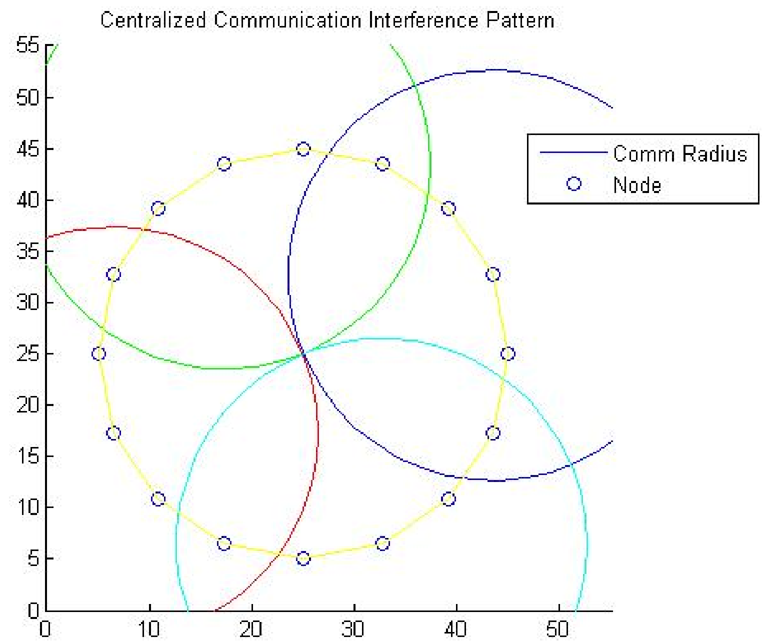

Figure B.1: Graphical Representation of Centralized Communication Interference

If there is only one channel available to the CRN, then each node must communicate back to the centralized node during one time interval. This means that N time steps are required. If two channels are available, then N/2 time steps are required. This trend continues with a lower bound of one required time step. It is assumed that there is one required time step for processing the collected data. The final result for the number of channels required for a given response time is shown in Equation (B.1).

$$c = \left\lceil \frac{N}{t-1} \right\rceil \tag{B.1}$$

Next, rearranging Equation (B.1), the equation for the time response given a bandwidth constraint, produces Equation (B.2).

$$t = \left\lceil \frac{N}{c} \right\rceil + 1 \tag{B.2}$$

B.2 Next Neighbor

In the case of Next Neighbor, nodes are arranged such that they can communicate with one another without interfering with other clusters. This allows all of the clusters to operate synchronously until the data is returned to the user. The clustering method also requires that a number of hops occur before a location can be made. Equation (B.3) is used to compute the time for the clusters to generate an estimate, where $t_{cluster}$ is the time for cluster estimation, t_{hops} is the time for the individual hops, and $t_{estimate}$ is a single time step for actual computation of the estimation.

$$\begin{aligned} t_{cluster} &= t_{hops} + t_{estimate} \\ &= n - 1 + 1 \\ &= n \end{aligned} \tag{B.3}$$

Next, the time required to return the packets to the central node is dependent on both the number of clusters and the number of channels available. This is given by Equation (B.4) and is similar to the Centralized results.

$$t_{return} = \left\lceil \frac{k}{c} \right\rceil \tag{B.4}$$

Finally, the two components from Equations (B.3) and (B.4) are added to yield the total time of the estimation, shown in Equation (B.5). This can be solved for bandwidth and yields Equation (B.6).

$$\begin{aligned} t &= t_{cluster} + t_{return} \\ &= n + \lceil \tfrac{k}{c} \rceil \end{aligned} \tag{B.5}$$

$$c = \left\lceil \frac{k}{t-n} \right\rceil \tag{B.6}$$

Figure B.2 shows the interference pattern for the Next Neighbor method. The small blue circles represent the intra-cluster communications; notice how they do not overlap between clusters. The large red circles represent the return information to the central node. These communications will interfere with one another.

B.3 Node Spacing

The Node Spacing Method creates the most complicated model for bandwidth usage. There are two cases: one where the clusters are large enough so that all the

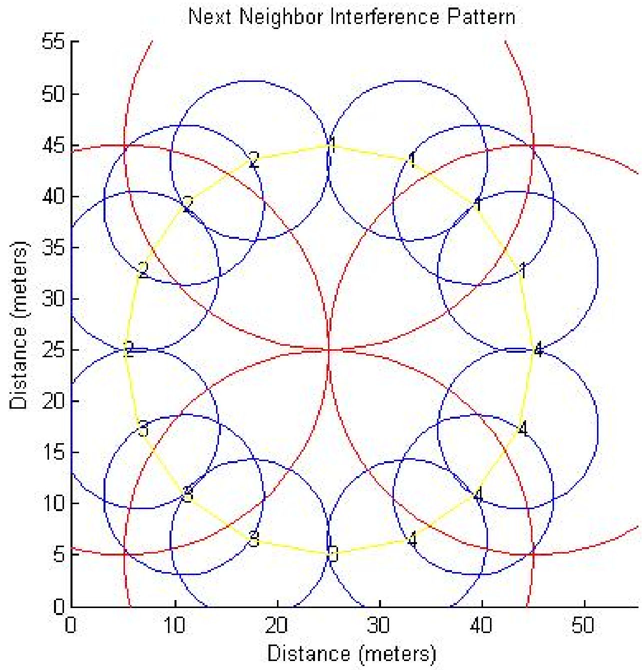

Figure B.2: Graphical Representation of Next Neighbor Communication Interference

clusters can operate in a synchronous fashion, or, if this is not the case, a second where the bandwidth must be computed using a more complex equation.

By using a MATLAB simulation, it was determined that the number of nodes required for all clusters to communicate synchronously on one channel was given by Equation (B.7). If Equation (B.7) holds true then, Equations (B.5) and (B.6) can be used to determine the Bandwidth and Time parameters.

$$N \geq (2k+1)k = 2k^2 + k \tag{B.7}$$

If N does not meet the criteria above, then the maximum number of clusters (k_{max}) that can communicate on the same channel is given by Equation (B.8). Again this was determined experimentally using a MATLAB simulation.

$$k_{max} = \lfloor n/2 \rfloor + 1 : N < 2k^2 + k \tag{B.8}$$

Using Equation B.8, t_{hops} can be computed, and the results are given by Equation (B.9)

$$t_{hops} = \left\lceil \frac{k}{k_{max} c}(n-1) \right\rceil + 1 : k_{max} c \leq k \tag{B.9}$$

Substituting Equation (B.9), into Equation (B.5), the final result for the Node Spacing method, is given by Equation (B.10), is obtained.

$$\begin{aligned} t &= t_{cluster} + t_{return} \\ &= \left\lceil \tfrac{k}{k_{max} c}(n-1) \right\rceil + 1 + \lceil \tfrac{k}{c} \rceil \end{aligned} \tag{B.10}$$

A graphical representation of the interference problems with Node Spacing is shown in Figures B.3 and B.4.

Notice that in Figure B.3, none of the rings overlap with any of the other rings. This means that there is no communication interference between the clusters, as long as the cluster initiators are wisely selected and spread out throughout the network. Cluster initiators should be placed as far apart from one another as possible to prevent interference. Also note in Figure B.4, that there are two circles that overlap, meaning

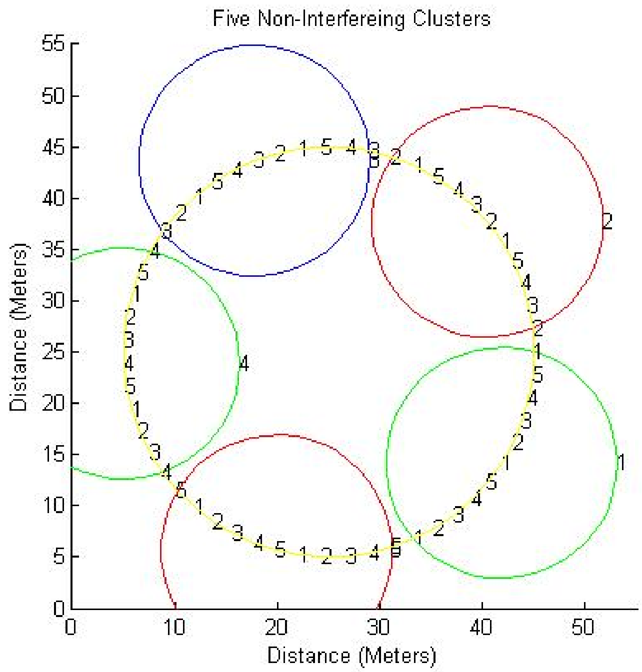

Figure B.3: No Communication Interference in Node Spacing

that their communication ranges will interfere with one another. Since only two of the circles overlap, all but one cluster may communicate at a time. This must be taken into account when setting up a CRN for localization purposes.

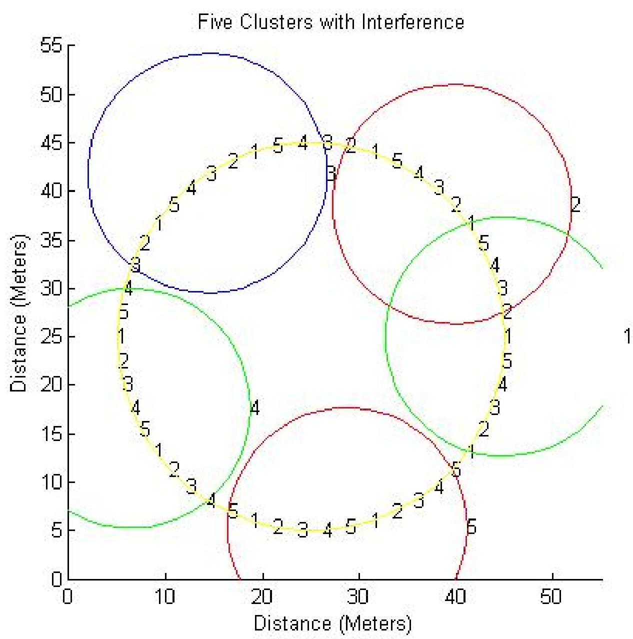

Figure B.4: Communication Interference in Node Spacing

Appendix C. Matlab Code and Discrete Event Simulation

This Appendix provides selected the Matlab scripts written for this research thesis. The focus is on the classes needed for Discrete Event Simulation (DES) and how the event system was established in this research effort. It also contains a selection of the code used in this research effort to explain some of the basic concepts.

C.1 Matlab DES

Matlab DES relies on event triggers. By setting up a class to throw an event whenever a trigger occurs, it is possible to have other classes listen and react to this trigger. The listening class must have a listener method, and it must be linked to the other class by listening for that event to occur. This is further explained in the following sections.

C.1.1 Handle vs Value Classes. First, it is important to understand the difference between Handle and Value Classes. A value class is a class, that once instantiated, it is stored as a variable. This means that if that variable is set equal to another variable, an exact copy of the same information is created. See Figure C.1.1 for an example. When a handle class is created, the variable referencing it is pointing at a memory location. If the variable is copied and either variable is altered, both variables will reflect the change.

It is important to understand that if either Var or Var2 from Figure C.1.1 are changed, there would be no impact on each other. If these variables were set up as

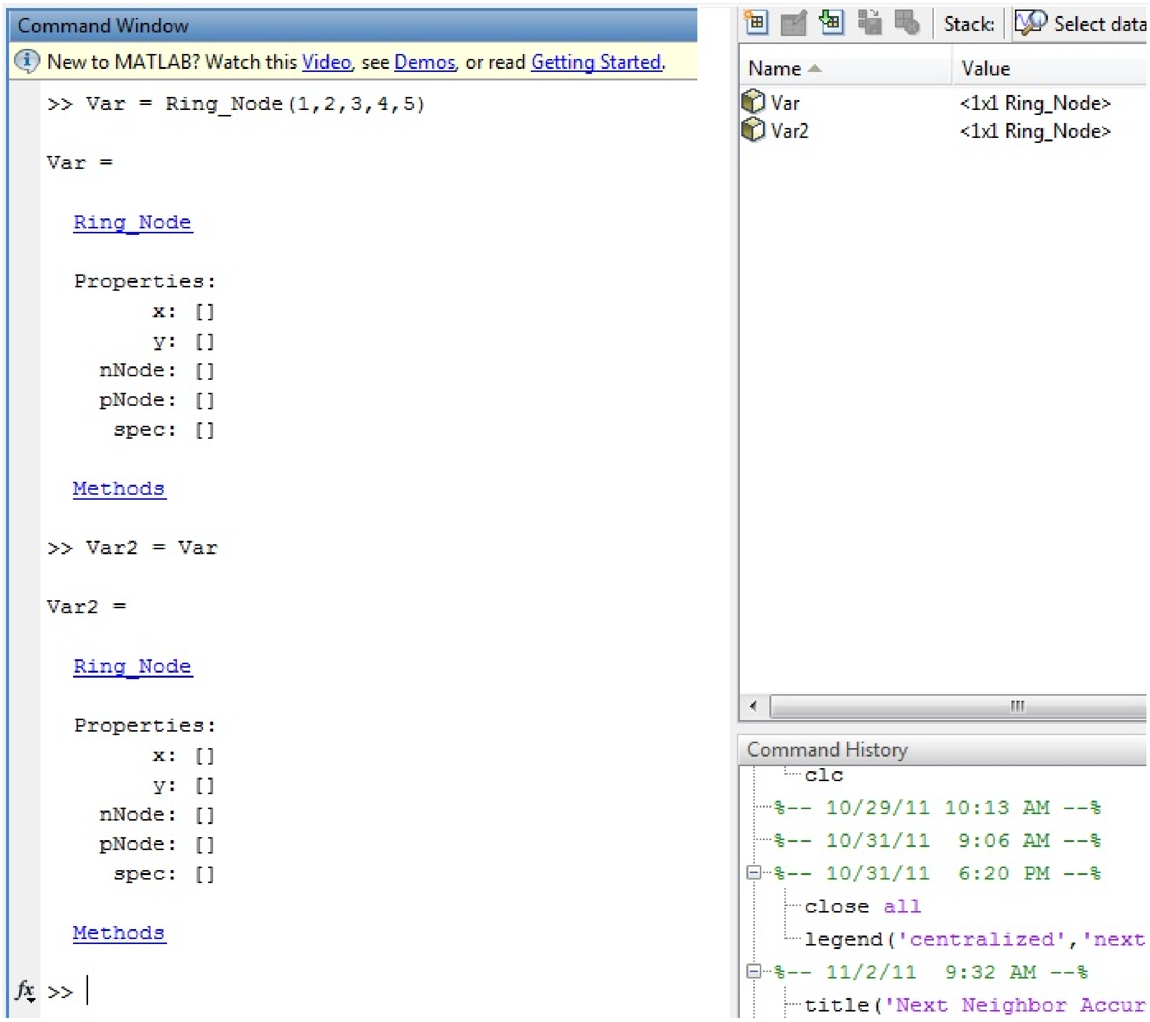

Figure C.1: Creation of Two Value Classes

handle classes, this would not be true. Because the handle class variables access a memory location, both variables reflect the changes made to one. The handle class is helpful, because it is easily passable from one class to another as a variable, and it can be influenced to affect every other reference to it. This is required to create an event that impacts multiple classes, such as a sensor node.

C.1.2 Matlab Event Handling. To create an event-driven simulation, an event class is created that controls which variables are passed from the event trigger

to the event listener. The event class used in this thesis is provided here in Listing C.1

```
classdef newSourceData < event.EventData
    %freq = frequency of detected souce
    properties
        freq = 0;
    end
    methods
        %when a source is detected tell someone
        function eventData = newSourceData(freq)
                eventData.freq = freq;
        end
    end
end
```

Listing C.1: Sample Event Class

This event class is used by creating an instance of it in the spectrum class. The Spectrum Class method that creates the event is shown in Listing C.2.

```
function addSource( self , source )
  if source.getFreq() < self.centerFreq + self.BW/2 || source...
      .getFreq() > self.centerFreq - self.BW/2
        self.sources = [self.sources source];
          notify(self, 'newSource', newSourceData(source....
              getFreq()))
      end
  end
```

Listing C.2: Event Creation

The notify statement is the event creation statement. First, it creates a newSourceData Class, which contains data about the frequency at which the source was detected. The notify statement creates the event with the label “newSource.” Any class that has a listener set up for the “newSource” event, will react and run the prescribed event handling method. The code for the event listener and the event handling method is shown in Listing C.3.

```
methods
  %Ring_Node Constructor
  function self = Ring_Node(locX,locY,ParentNode, ...
      minimum_to_locate, IP_Address,...
                                             spect, initiate, ...
                                                available_battery)
     ...
         self.lh = addlistener(self.spec,'newSource',@self....
             newSourceDetection);
         ...
        end
   ...
  %called if a new source enters the spectrum
     function newSourceDetection(self, eventSrc, eventData)
        if( self.init == 1)
           self.Locate([],[], eventData.freq, self.name, 0)
        end
     end
     ...
end
```

Listing C.3: Event Listener and Handling method

The Listener is listening for a "newSource" event to occur. Once it does, the newSourceDetection class is called. In this particular usage, the method checks if this sensor is an initializer; if it is, it will begin attempting to locate the transmitter. If it is not an initializer, the node simply waits for a packet transmission from another node. Figure C.2 shows that the listener is created by a ring node object and the ring node object is referenced when an event is thrown.

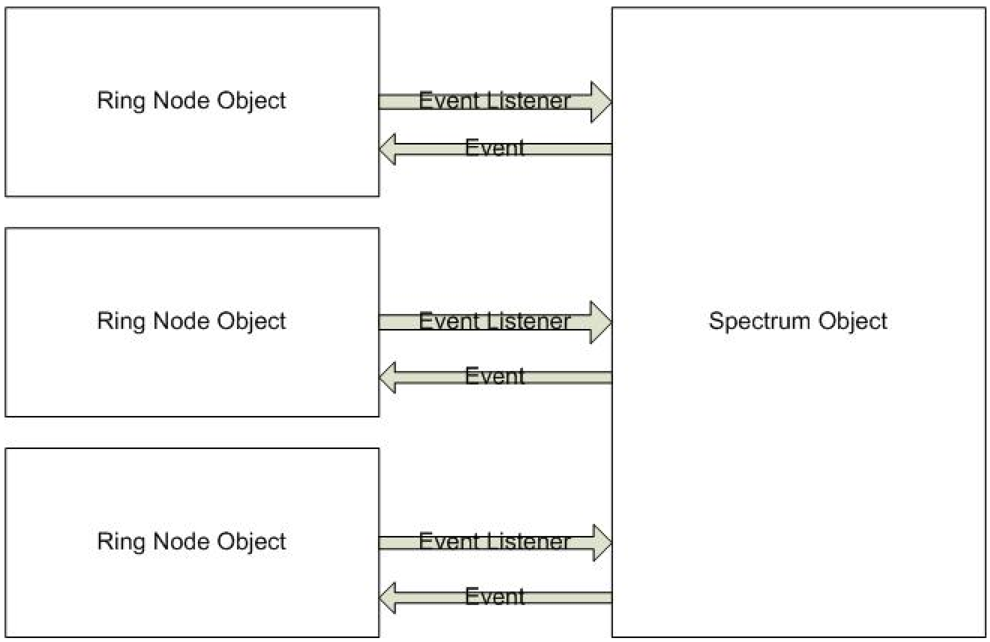

Figure C.2: Event Handling in Matlab

Appendix D. Validation Feasibility Test at CORNET

The purpose of this appendix is to explain the methodology and experimental results, used on the CORNET, to determine if the network is a feasible location for distributed RSS experiments.

D.1 Data Collection on Cornet at Virginia Tech

CORNET has 48 Universal Software Radio Peripherals 2 (USRP2) with custom daughter boards based on the Motorola RFIC4. The nodes span four floors of the ICTAS building on the Virgina Tech campus and are connected to blade servers on the building's first floor. The nodes can be accessed by logging into each node using the Secure Shell (SSH) protocol. After logging in, the USRP2 nodes can be controlled using GNU Radio scripts. GNU Radio scripts were generated using GNU Radio Companion (GRC), which is a graphical user interface for developing GNU Radio flow diagrams. Several of the nodes were malfunctioning at the time of the experiments. For this reason, only the first floor was used for the experiments. The layout of CORNET can be seen in Figure D.1. Figure D.2 shows the actual interior pictures of the building, the nodes are located above the dropped ceiling tiles.

Two experiments were performed on the test bed. The first was a benchmark test to gain understanding about how well the nodes could communicate with one another. Using the Benchmark Tx and Benchmark Rx files included in GNU Radio, one node was set to transmit while all other nodes where set to receive. The nodes were set to transmit and receive on the 465 MHz band. The frequency band was

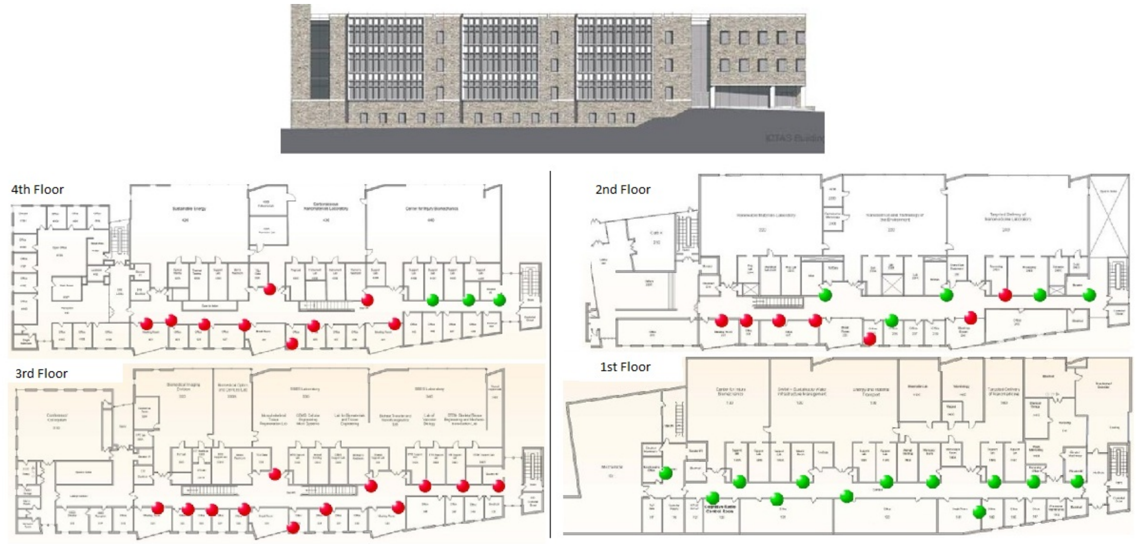

Figure D.1: CORNET Building Layout

selected after previous experiments showed unmanageable interference at the other frequencies. This frequency band also closely emulates the frequency that is used by Family Radio Service (FRS) walkie-talkies. All nodes were used as transmitters in different iterations of the experiment. Two statistics were monitored: packets received and packets received correct. Each trial transmitted 667 packets, which were used to compute the percentage of packets received and percentage of packets received correctly.

The second experiment performed an energy-based localization estimate using RSS. Each node transmitted on four different tone frequencies: 464.930 MHz, 464.965 MHz, 465.035 MHz, 465.070 MHz. Again these frequencies where selected because of their similarities to FRS channels. The center oscillator in each case was set to 465 MHz. This means that the tones where spaced slightly to the left or right of the center oscillator. Each node transmitted on this frequency while the other nodes listened

Figure D.2: CORNET Interior

for the tone for five seconds. The data was then collected and processed for RSS measurements. The RSS measurements can be placed into a localization algorithm using the location of the receiving nodes and the RSS values collected. The RSS measurements will then generate the final position estimation.

D.2 Data Processing and Validation of Model

Once the data, from the CORNET, has been obtained it is then processed using MATLAB to take the Fourier transform, locate energy spikes at the expected frequency, then integrate over the signal bandwidth. In this case the bandwidth is based on the potential error between each USRP2's center oscillator. After the RSS measurements were computed, the data was put into the RSS localization script

for both the centralized and distributed cases. Accuracy results were then compared between the two. There are concerns of error being introduced, because of the extreme multipaths exhibited in the CORNET test bed, created by piping and duct work in the ceiling of each floor. Additional multipath modeling may need to be performed in order to fully understand the final accuracy results of each localization method on the CORNET testbed.

D.3 Results

Figure D.3 shows a sample of the results from the experiments run on CORNET at Virginia Tech. It can be seen in this result that an energy spike was generated and detected by another sensor in the network. This spike was not found consistently, which can be explained by the results shown in Figure D.3.

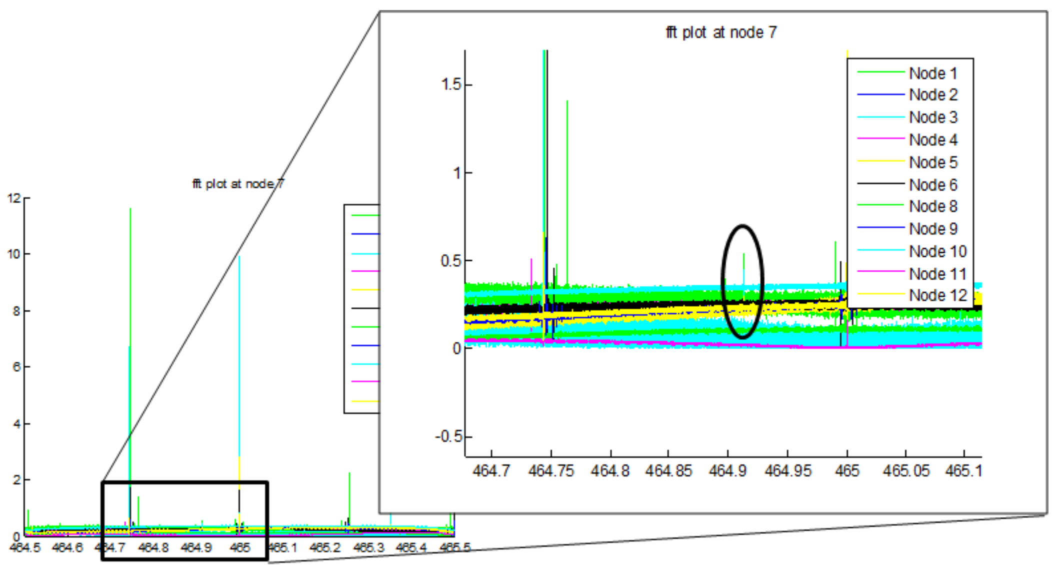

Figure D.3: Sample Result from the CORNET experiments

To explain the successful detection of the source at some nodes, and the failure to detect at others, an analysis of the packet reception rate was performed. First, for each possible transmitter, a series of packets were transmitted. Then, the number of received packets and the number of correctly received packets were computed as a percentage of total packets transmitted. Next, the collected results were then plotted against the distance between the nodes on the CORNET layout. Finally, these results are shown in Figures D.4 and D.5.

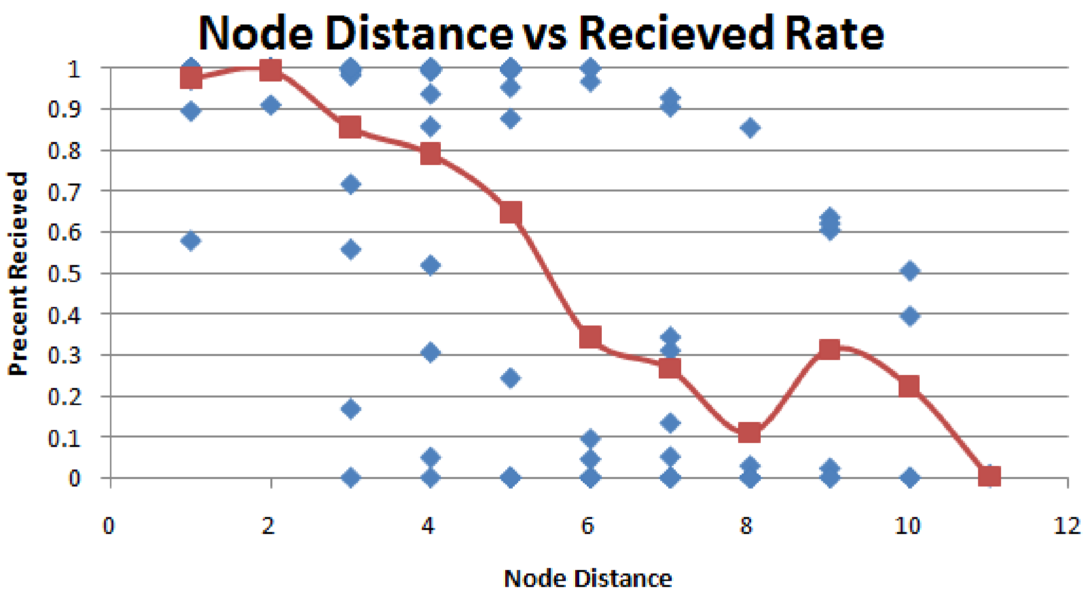

Figure D.4: Successfully Received Nodes Based on Distance Between Nodes

The line in Figures D.4 and D.5 indicates the average correct packet reception. The trend is downward as the distance grows between nodes, but is not monotonic. It should also be noted that early in both methods of packet analysis, there are nodes that received zero packets from the transmitting source. This is likely caused by some nodes being in adjacent rooms and not in the central hallway, as well as the issue

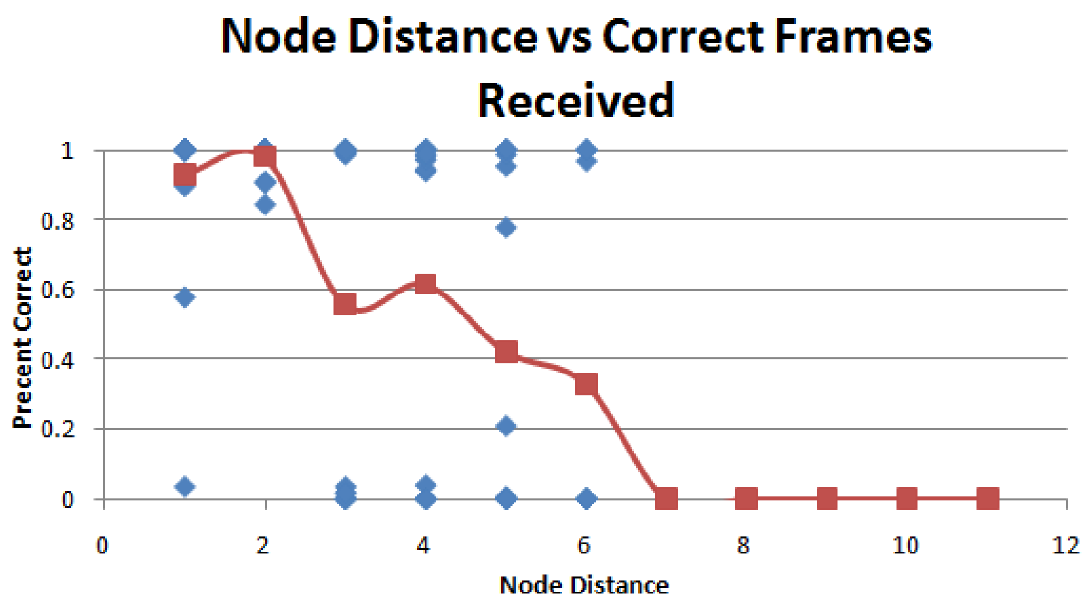

Figure D.5: Correctly Received Packets Based on the Distance Between Nodes

with duct work and piping located in the ceiling of the hallway. The nodes are also placed in the ceiling and caused several issues with multipath, and they did not allow for any LOS communications. It will be challenging to generate accurate results from this test bed.

This Appendix provides initial results discovered for the future works that are currently in progress.

E.1 Changing the Communication Graph

The first results show the power problems that occur with varying connectivity. Figure E.2 shows the power usage per node. The dotted line represents the centralized power usage determined previously. The lines with circles represent the power usage from the node spacing method and the solid line represents the next neighbor method. The blue is the average power usage. 36 nodes were used with 12 clusters. The *Node Range* is a method of explaining the communication range of each node. A node range of one represents only being able to reach the next node in the network. A node range of two means it can skip the over one node in the network and reach the node that is two nodes away. This continues until a fully connected network is achieved.

One interesting aspect of this plot, is that the next neighbor method has no variance in the required power. Regardless of the communication range, the clustering method only uses nearby neighbors. After 12 node range, Node Spacing becomes a similar flat line to the Next Neighbor method results. This is because at 12 Node range, the network is "connected enough," and all the links are now used. An "erratic" growth pattern is created by nodes that use multiple times to relay packets to other nodes in the network. One limitation to this system is it only uses one route to reach another node in the cluster. Using multiple routes may better spread the power usage

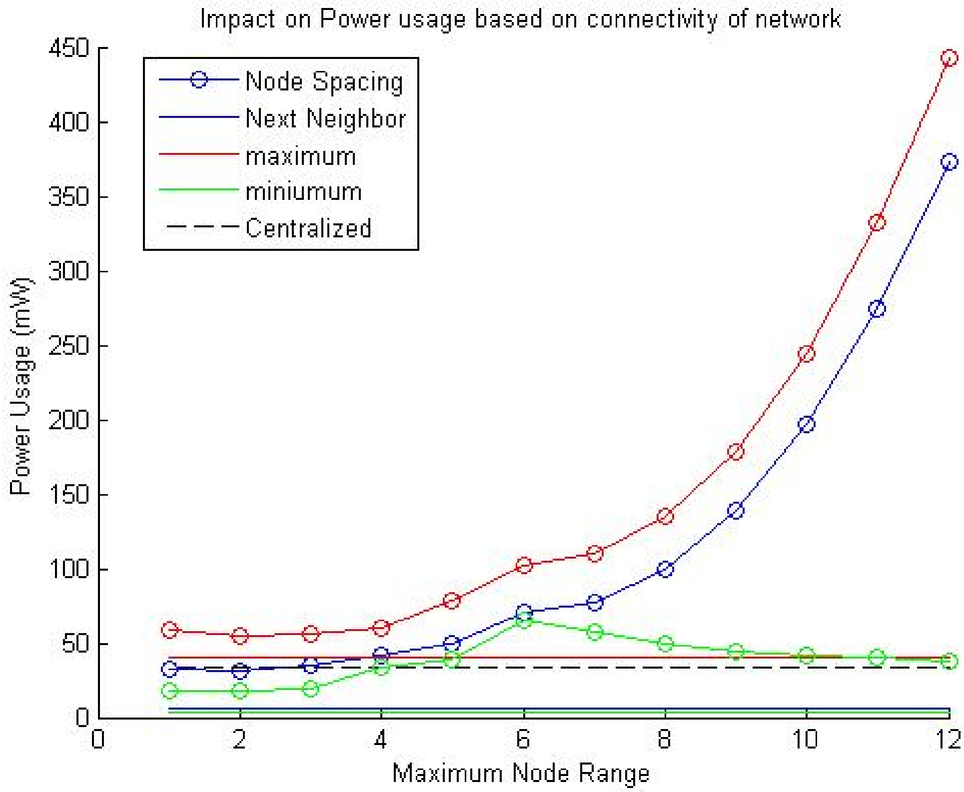

Figure E.1: Power Results created by varying the communication graph

out in the network. One interesting discovery is that using only the shortest link reduces the overall power usage of the system for the Node Spacing method.

Next, the impact on latency was explored using the variable connectivity. The centralized was marked for four channels available for communication. There are no collisions and no waiting for a channel in the distributed cases for simplicity at this point. In this case, because of the increased packet flow, it becomes more important to take into account collisions. Figure E.2 shows the case with 36 nodes and 12 clusters. With only a node distance of one node, there is a large delay in time steps compared to both centralized and distributed methods. As the number of reachable nodes increases, the Node Spacing method approaches the Next Neighbor, since the

number of hops needed for each communication approaches one. Next Neighbor is again unaffected, because of its design.

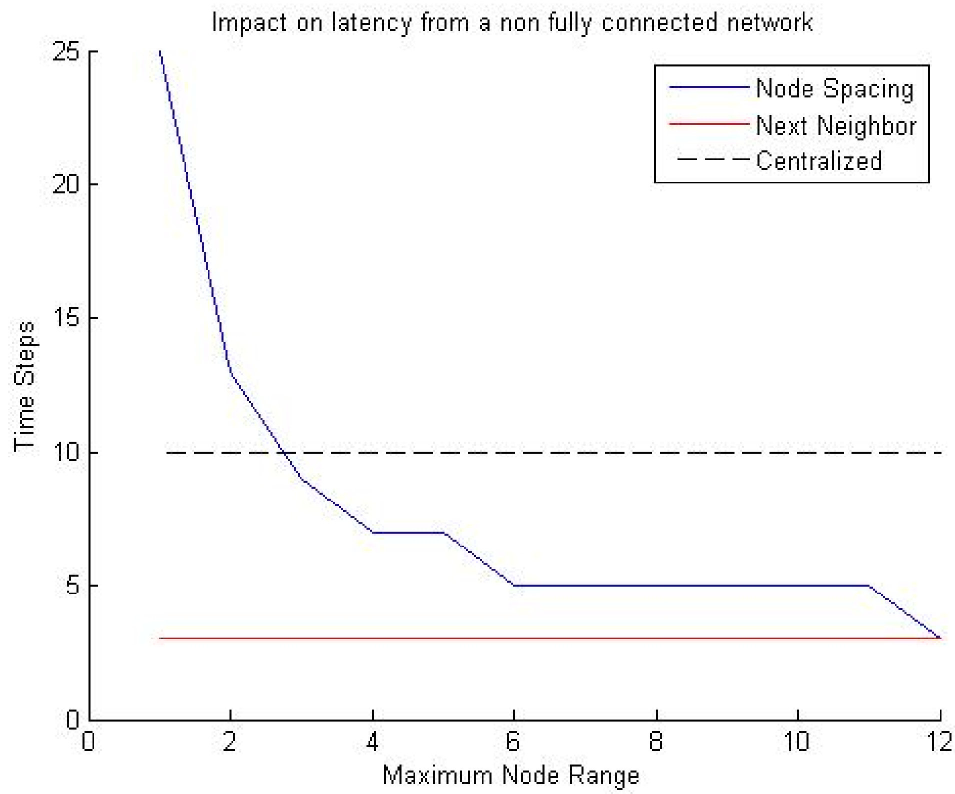

Figure E.2: Latency Results created by varying the communication graph

E.2 Introducing Packet Loss

One situation that may occur if distributed localization techniques are employed is a packet loss. In the event of a packet loss, an entire clusters RSS measurements and its cluster's estimate is lost. This clearly negatively impacts the performance of the system. In the worst case scenario, no estimates are made, because all of the clusters dropped a packet. In the future, additional fault tolerance must be added to the clustering communication protocols, such as acknowledgments. A simulation

was run using 36 nodes and 4 clusters, with a 25% chance that a cluster would drop a packet. The results are shown in figure E.3. There is some decrease in accuacy particulaly inside the sensor ring. There were also several instances where no target source was detected by the system.

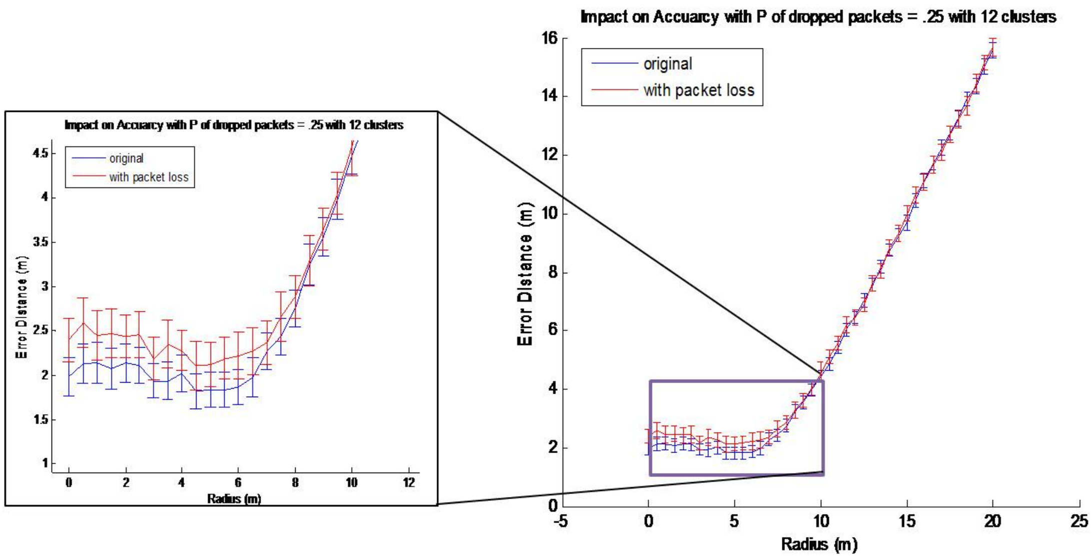

Figure E.3: Impact of Dropping a Packet with 25% Chance of a Cluster Dropping a Packet

E.3 Grid Topology

The grid topology is a common topology for experiments using RSS localization. To continue to validate the Node Spacing and Next Neighbor distributed methods, it is important to apply them to the grid topology. Again, the focus is on spreading out the nodes in the case of Node Spacing, and to group nodes tightly with Next Neighbor. An example of how to cluster Node Spacing is shown in Figure E.4; Next Neighbor is shown in Figure E.5.

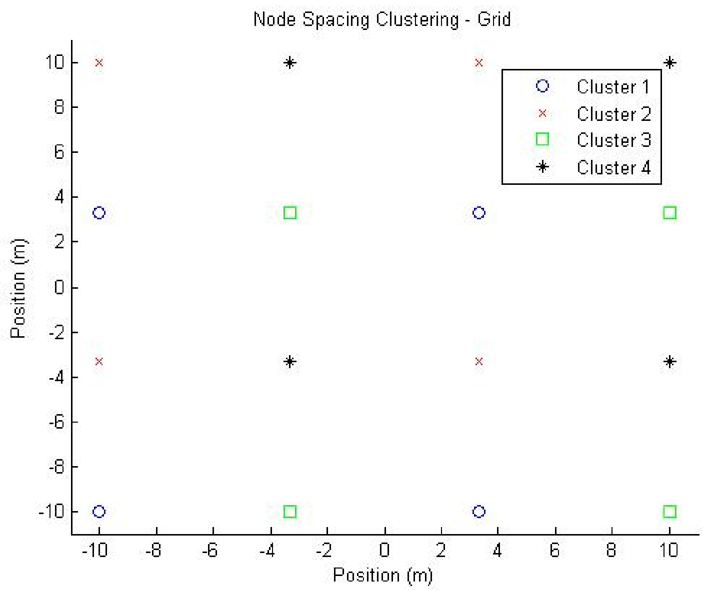

Figure E.4: Node Spacing Clustering with 16 Nodes in a Grid

Using the clustering shown in Figures E.4 and E.5 an initial simulation was run testing these layouts' accuracy. To test the grid layout sensors were placed using rectangular coordinates, (x, y). The experiment configuration included sensors placed from (0m, 0m) to (20m, 20m) with locations every two meters in each direction. Figure E.6 shows the target positions, the sensor positions, and the search space. Only one quadrant is explored, because of the symmetry of the grid.

The accuracy at each location was measured and then plotted showing the regions of varying Error Distances. These plots are shown in Figures E.7, E.8 and E.9.

After reviewing each of the grid's accuracy figures, the Centralized technique is the most accurate. This is because the nodes are equally spaced throughout the entire grid, instead of simply around it. The additional spacing of nodes allows for

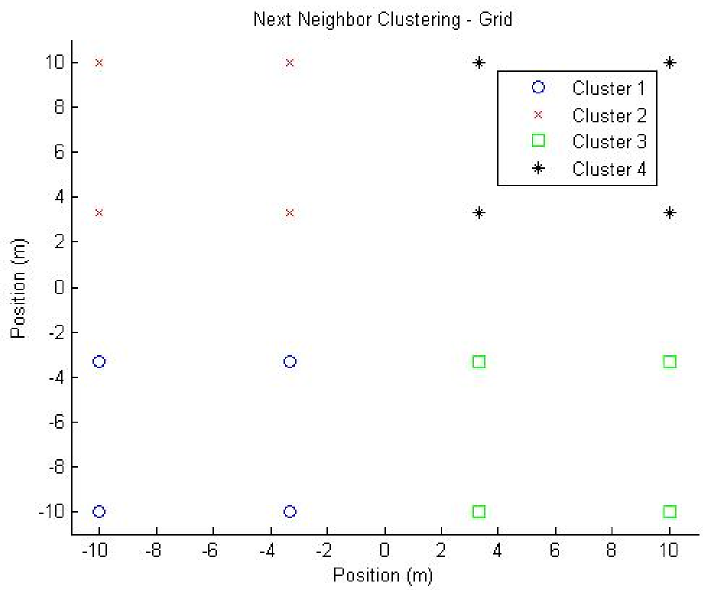

Figure E.5: Next Neighbor Clustering with 16 Nodes in a Grid

greater difference in recorded power measurement form each node. Node Spacing is the most accurate of the distributed cases, and provides reasonable accuracy inside the sensor network. Next Neighbor is the least accurate.

E.4 Conclusions

In this Appendix, three initial experiments were explored. The first involved making changes to the communication graph of the sensor network. It was found that reducing node range had no impact on the Next Neighbor technique, but reduced the overall power usage of the Node Spacing method. The Node Spacing method incurred additional latency, approximately 600% more, when the node range was set to one. The second experiment, packet loss, examined what happened when packets were dropped in the network. It was determined that the loss of packets reduced the accuracy of the system, because each time a packet was lost, one cluster was unable to

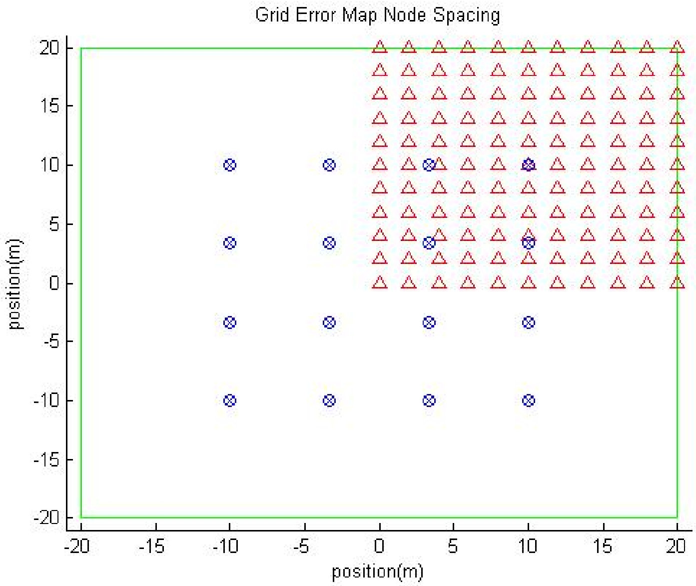

Figure E.6: Search Space Map for Grid Experiment

make an estimate. To correct this problem, additional fault tolerance must be explored and applied to the distributed systems. Finally, applying all three clustering methods to a grid topology was explored. It was discovered that the additional spacing of each sensor improved the overall accuracy for each method. The Centralized method was shown to be the most accurate, both inside and outside the grid. The Node Spacing technique was the second most accurate, and the Next Neighbor method was the least accurate. To continue this research, additional grid sizes should be applied, as well as an analysis of the power and latency used for each of the topologies.

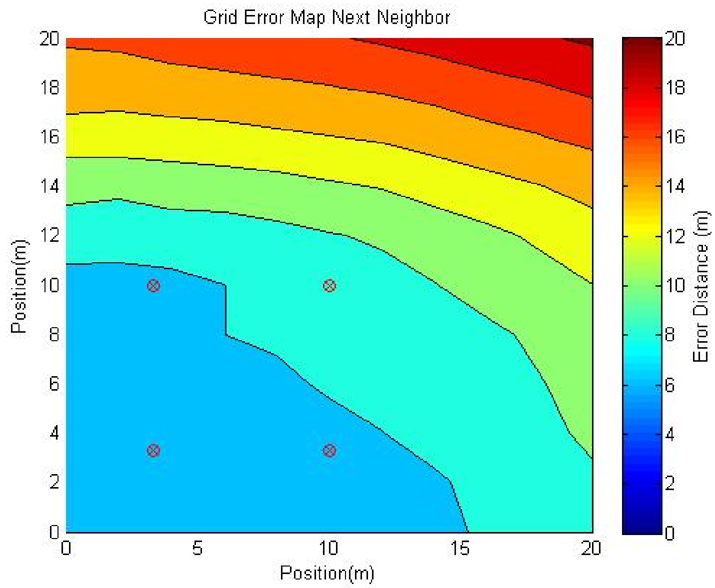

Figure E.7: Grid Accuracy for Next Neighbor

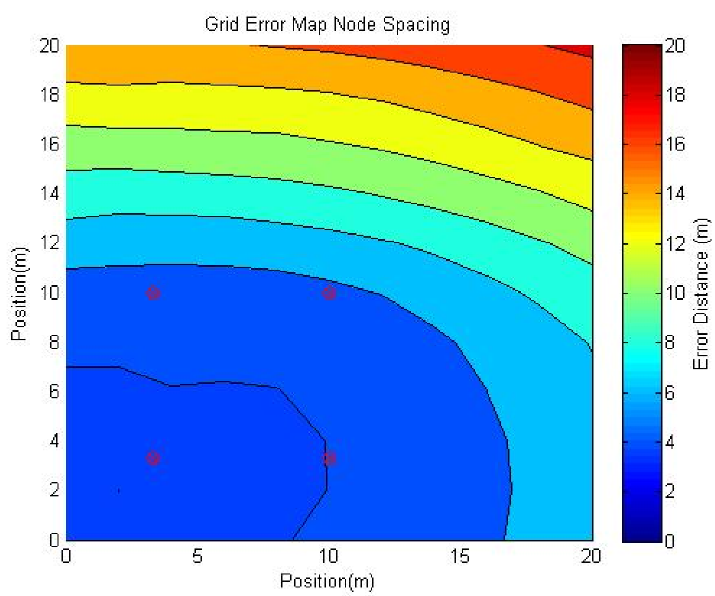

Figure E.8: Grid Accuracy for Node Spacing

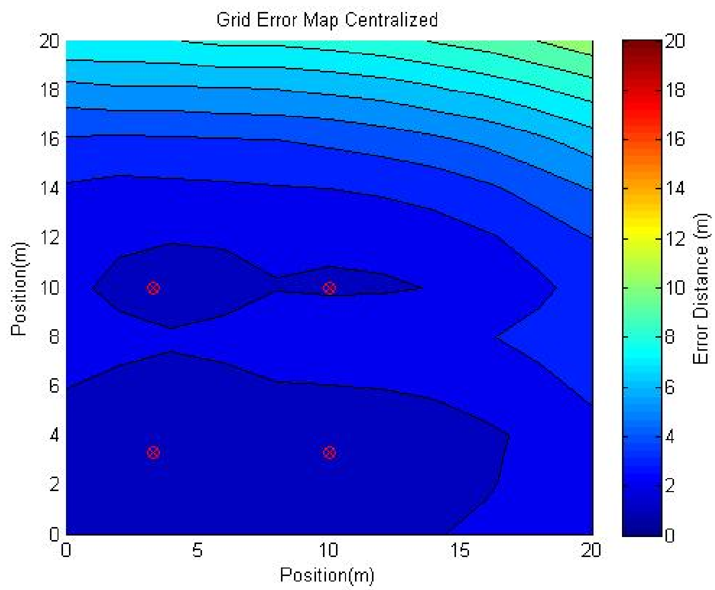

Figure E.9: Grid Accuracy for Centralized

Bibliography

1. D. Datla, X. Chen, T. Tsou, S. Raghunandan, S. M. Hasan, H. Reed, B. Fette, C. B. Dietrich, J. Kim, and T. Bose, “Wireless distributed computing: A survey of research challenges,” *Communications Magazine*, vol. 50, no. 1, pp. 144–152, January 2012.
2. R. K. Martin, C. Anderson, R. Thomas, and A. King, “Performance analysis of 3D radio tomographic imaging,” *IEEE Signal Processing Letters*, vol. Dec, 2011.
3. R. K. Martin and R. Thomas, “Algorithms and bounds for estimating loction, directionality, and environmental parameters of primary spectrum users,” *IEEE Transactions On Wireless Communications*, vol. 8, pp. 1–10, 2009.
4. W. J. A. Dahm, *Technology Horizons*, May 2010, vol. 1.
5. A. H. Cordesman, *The Iraq War: strategy, tactics, and military lessons*, P. A. J. Bacevich, Ed. Center for Strategic International Studies, 2003.
6. R. B. Bacchus, A. J. Fertner, C. S. Hood, and D. A. Robertson, “Long-term, wide-band spectral monitoring in support of dynamic spectrum access networks at the IIT spectrum observatory,” in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2008, pp. 1–10.
7. A. Sampath, H. Dai, H. Zheng, and B. Y. Zhao, “Multi-channel jamming attacks using cognitive radios,” in *Proceedings of 16th International Conference on Computer Communications and Networks*, 2007, pp. 352–357.
8. G. Sun, J. Chen, W. Guo, and K. J. R. Liu, “Signal processing techniques in network aided positioning,” *IEEE Signal Processing Magazine*, vol. 22, pp. 12–23, 2005.
9. N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: Cooperative localization in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 22, pp. 54–69, 2005.
10. D. Datla, H. L. Volos, S. Hasan, J. H. Reed, and T. Bose, “Wireless distributed computing in cognitive radio networks,” *Elsevier Ad Hoc Networks Journal*, vol. Speical Issue on Cognitive Radio Ad Hoc Networks, pp. 1–13, 2011.
11. G. Tel, *Introduction to Distributed Algorithms*, P. Vitanyi, Ed. Cambridge Unitversity Press, 2000.
12. M. D. Silvius, “Building a dynamic spectrum access smart radio with application to public safety disaster communications,” Ph.D. dissertation, Virginia Tech, 2009.

13. W. Lee and I. F. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Transacations On Wireless Communications*, vol. 7, pp. 3845–3845, 2008.

14. D. Datla, S. M. Hasan, J. H. Reed, and T. Bose, "Fundamental issues of wireless distributed computing in SDR networks," in *Wireless Innovation Forum Technical Conference*, Dec 2010.

15. Y. Yu and V. K. Prasanna, "Exploring energy-latency tradeoffs for data gathering in wireless sensor netowrks," *IEEE Transactions on Wireless Communications*, vol. 5, pp. 7–11, Nov 2006.

16. J. Acimovic, B. Beferull-Lozano, and R. Cristescu, "Adaptive distributed algorithms for power-efficient data gathering in sensor networks," in *International Conference on Wireless Networks, Communications and Mobile Computing*, 2005.

17. R. Abdolee and B. Champagne, "Distributed blind adaptive algorithms based on constant modulus for wireless sensor networks," in *6th International Conference on Wireless and Mobile Communications*, 2010, pp. 303–308.

18. X. Pi and H. Yu, "A distributed and cooperative target localization algorithm in wireless sensor networks," in *Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies.*, Dec. 2005, pp. 887 – 889.

19. H. Sabaghian-Bidgoli, N. Yazdani, and F. Lahouti, "A distributed algorithm for node localization by connectivity in large scale wireless mesh sensor networks," in *14th International CSI Computer Conference (CSICC)*, 2009.

20. M. Berthold and D. J. Hand, Eds., *Intelligent Data Analysis.* Springer, 2003.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-03-2012	Master's Thesis	Sept 2010 - Mar 2012

4. TITLE AND SUBTITLE Distributed Localization of Active Transmitters in a Wireless Sensor Network	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Vincent, Oba L, 2d Lt	**5d. PROJECT NUMBER** 11G182
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way Wright-Patterson AFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GE/ENG/12-41
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Labroatory Sensors Directorate Dr. Vasu Chakravarthy 2241 Avionic Circle WPAFB OH, 45433 vasu.chakravarthy@wpafb.us.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RYWE **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In today's military environment, emphasis has been placed on bandwidth efficiency and total use of the available spectrum. Current communication standards divide the spectrum into several different frequency bands, all of which are assigned to one or multiple primary users. Cognitive Radio utilizes potential white spaces that exist between currently defined channels or in time. One under-explored dimension of white space exploration is spatial. If a frequency band is being used in one region, it may be underutilized, or not occupied in another. Using an active localization method can allow for the discovery of spatial white; trying to spatially map all of the frequencies in a large area would become very computationally intensive, and may even be impractical using modern centralized methods. Applying a distributed method and the concepts discussed in Wireless Distributed Computing to the problem can be scaled onto many small wireless sensors and could improve the measuring system's effectiveness. For a bandwidth contested environment that must be spectrally mapped, three metrics stand out: Accuracy, Power Consumption, and Latency

15. SUBJECT TERMS

Localization, Wireless Distributed Computing, Wireless Sensor Networks, Received Signal Strength

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Mark D. Silvius USAF
U	U	U	UU	149	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x 4684 mark.silvius@afit.edu